

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2001 年 11 月 1 日 (01.11.2001)

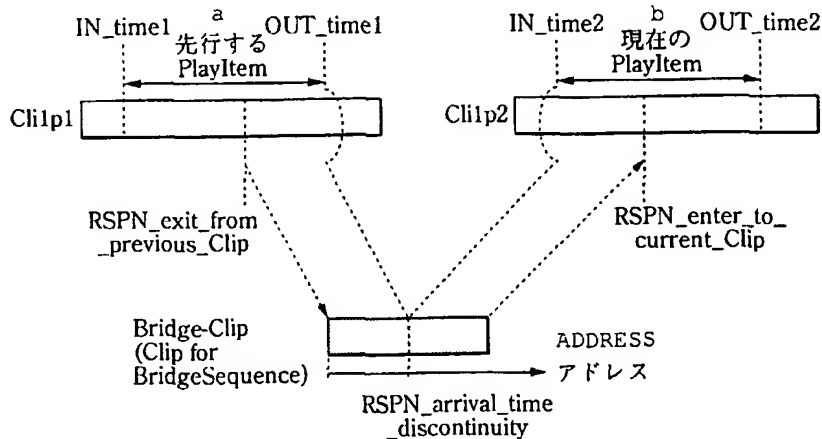
PCT

(10) 国際公開番号
WO 01/82610 A1

- (51) 国際特許分類⁷: H04N 5/93, G11B 20/10 Motoki) [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP];
〒141-0001 東京都品川区北品川6丁目7番35号 ソニー
(21) 国際出願番号: PCT/JP01/03417 株式会社内 Tokyo (JP).
(22) 国際出願日: 2001 年 4 月 20 日 (20.04.2001) (74) 代理人: 小池 晃, 外 (KOIKE, Akira et al.); 〒105-
0001 東京都港区虎ノ門二丁目6番4号 第11森ビル
(25) 国際出願の言語: 日本語 Tokyo (JP).
(26) 国際公開の言語: 日本語 (81) 指定国 (国内): CN, KR, US.
(30) 優先権データ: (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE,
特願2000-183769 2000 年 4 月 21 日 (21.04.2000) JP DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
特願2000-271550 2000 年 9 月 7 日 (07.09.2000) JP
(71) 出願人 (米国を除く全ての指定国について): ソニー株
— 添付公開書類:
株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001
— 国際調査報告書
東京都品川区北品川6丁目7番35号 Tokyo (JP).
— 補正書
(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 加藤元樹 (KATO, 2 文字コード及び他の略語については、定期発行される
各PCTガゼットの巻頭に掲載されている「コードと略語
のガイダンスノート」を参照。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体



a...PREVIOUS PlayItem
b...CURRENT PlayItem

(57) Abstract: When an instruction to continuously reproduce first and second AV stream is issued, a third AV stream comprising a predetermined portion of the first AV stream and a predetermined portion of the second AV stream is created and reproduced when the reproduction of the first AV stream is changed to the reproduction of the second AV stream. Address information concerning the third AV stream and including information on the address of the source packet of the first AV stream at the timing when the reproduction of the first AV stream is switched to the reproduction of the third AV stream and information on the address of the source packet of the second AV stream at the timing when the reproduction of the third AV stream is switched to the reproduction of the second AV stream is created. Hence the continuity of reproduction of separately recorded AV streams is maintained.

[続葉有]

BEST AVAILABLE COPY

WO 01/82610 A1



(57) 要約:

第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示されたとき、第1のAVストリームの所定部分と第2のAVストリームの所定部分からなる第3のAVストリームが生成され、第3のAVストリームは、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される。第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングの第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングの第2のAVストリームのソースパケットのアドレスの情報とからなるアドレス情報を生成する。これにより、別々に記録されたAVストリームの連続性を保つように再生できる。

明細書

情報処理装置及び方法、プログラム、並びに記録媒体

技術分野

本発明は、情報処理装置及び方法、プログラム、並びに記録媒体に関し、特に、再生区間における動画像の連続性を保つ情報処理装置及び方法、プログラム、並びに記録媒体に関する。

背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース(供給源)としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group)2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。或いは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを1度デ

コードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。但し、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要がある。

また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

上述したような記録装置において記録媒体に記録されたデータを再生する際、所定のピクチャまで再生し、そのピクチャから時間的に離れた位置に位置するピクチャを続けて再生するといった、いわゆるスキップ再生というものがある。スキップ再生を行った際、再生する映像に時間的な連続性が途切れてしまことがあるといった課題があった。

発明の開示

本発明の目的は、このような状況に鑑みて、再生区間における動画像の連続性を保つように再生できるようにすることにある。

本発明に係る情報処理装置は、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の

部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成手段と、生成手段により生成された第3のAVストリームとアドレス情報を記録する記録手段とを含む。

生成手段により生成されたアドレス情報に含まれる第1のAVストリームのソースパケットのアライバルタイムスタンプと、第3のAVストリームの最初に位置するソースパケットのアライバルタイムスタンプは連続しており、且つ、生成手段により生成されたアドレス情報に含まれる第2のAVストリームのソースパケットのアライバルタイムスタンプと、第3のAVストリームの最後に位置するソースパケットのアライバルタイムスタンプは連続しているようにすることができる。

第3のAVストリーム内のソースパケットのアライバルタイムスタンプには、ただ1つの不連続点が存在するようにすることができる。

生成手段により生成されたアドレス情報に含まれる第1のAVストリームのソースパケットのアドレスの情報で示されるソースパケット以前のAVストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、アドレスは決定されるようにすることができる。

生成手段により生成されたアドレス情報に含まれる第2のAVストリームのソースパケットのアドレスの情報で示されるソースパケット以後のAVストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、アドレスは決定されるようにすることができる。

第3のAVストリームが記録媒体上で所定の大きさ以上の連続領域に配置されるように、第3のAVストリームが生成されるようにすることができる。

本発明に係る情報処理方法は、第1のAVストリームから第2のAVストリー

ムへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含む。

本発明に係る記録媒体のプログラムは、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含む。

本発明に係るプログラムは、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップをコンピュータに実行させる。

本発明に係る情報処理装置は、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームを記録媒体から読み出す第1の読出手段と、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を記録媒体から読み出す第2の読出手段と、第2の読出手段により読み出された第3のAVストリームに関連する情報に基づいて、第1の読出手段により読み出された第1のAVストリームから第3のAVストリームへ再生を切り換え、第3のAVストリームから第2のAVストリームへ再生を切り換えて再生する再生手段とを含む。

本発明に係る情報処理方法は、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームの記録媒体からの読み出しを制御する第1の読出制御ステップと、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第2の読出制御ステップと、第2の読出制御ステップの処理で読み出しが制御された第3のAVストリームに関連する情報に基づいて、第1の読出制御ステップの処理で読み出しが制御された第1のAVストリームから第3のAVストリームへ再生を切り換え、第3のAVストリームから第2のAVストリームへ再生を切り換えて再生する再生ステップとを含むことを特徴とする。

本発明に係る記録媒体のプログラムは、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームの記録媒体からの読み出しを制御する第1の読出制御ステップと、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVスト

リームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第2の読出制御ステップと、第2の読出制御ステップの処理で読み出しが制御された第3のAVストリームに関連する情報に基づいて、第1の読出制御ステップの処理で読み出しが制御された第1のAVストリームから第3のAVストリームへ再生を切り換え、第3のAVストリームから第2のAVストリームへ再生を切り換えて再生する再生ステップとを含む。

本発明に係るプログラムは、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームの記録媒体からの読み出しを制御する第1の読出制御ステップと、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第2の読出制御ステップと、第2の読出制御ステップの処理で読み出しが制御された第3のAVストリームに関連する情報に基づいて、第1の読出制御ステップの処理で読み出しが制御された第1のAVストリームから第3のAVストリームへ再生を切り換え、第3のAVストリームから第2のAVストリームへ再生を切り換えて再生する再生ステップとをコンピュータに実行させる。

本発明に係る記録媒体は、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームと、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録されている。

本発明に係る情報処理装置及び方法、並びにプログラムにおいては、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームが生成されると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が生成される。

本発明に係る情報処理装置及び方法、並びにプログラムにおいては、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームが記録媒体から読み出され、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録媒体から読み出され、読み出された第3のAVストリームに関連する情報に基づいて第1のAVストリームから第3のAVストリームへ再生が切り換えられ、第3のAVストリームから第2のAVストリームへ再生が切り換えられて再生される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の構成を示す図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlayListとVirtual PlayListについて説明する図である。

図 4 A～図 4 C は、Real PlayListの作成について説明する図である。

図 5 A～図 5 C は、Real PlayListの削除について説明する図である。

図 6 A 及び図 6 B は、アSEMBル編集について説明する図である。

図 7 は、Virtual PlayListにサブバスを設ける場合について説明する図である。

図 8 は、PlayListの再生順序の変更について説明する図である。

図 9 は、PlayList上のマークとClip上のマークについて説明する図である。

図 1 0 は、メニューサムネイルについて説明する図である。

図 1 1 は、PlayListに付加されるマークについて説明する図である。

図 1 2 は、クリップに付加されるマークについて説明する図である。

図 1 3 は、PlayList、Clip、サムネイルファイルの関係について説明する図である。

図 1 4 は、ディレクトリ構造について説明する図である。

図 1 5 は、info.dvrのシンタクスを示す図である。

図 1 6 は、DVR volumeのシンタクスを示す図である。

図 1 7 は、Resumevolumeのシンタクスを示す図である。

図 1 8 は、UIAppInfovolumeのシンタクスを示す図である。

図 1 9 は、Character set valueのテーブルを示す図である。

図 2 0 は、TableOfPlayListのシンタクスを示す図である。

図 2 1 は、TableOfPlayListの他のシンタクスを示す図である。

図 2 2 は、MakersPrivateDataのシンタクスを示す図である。

図 2 3 は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図 2 4 A～図 2 4 C は、PlayListについて説明する図である。

図 2 5 は、PlayListのシンタクスを示す図である。

図 2 6 は、PlayList_typeのテーブルを示す図である。

図 2 7 は、UIAppinfoPlayListのシンタクスを示す図である。

図 2 8 A～図 2 8 C は、図 2 7 に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

図 2 9 は、PlayItemについて説明する図である。

図 3 0 は、PlayItemについて説明する図である。

- 図 3 1 は、PlayItemについて説明する図である。
- 図 3 2 は、PlayItemのシンタクスを示す図である。
- 図 3 3 は、IN_timeについて説明する図である。
- 図 3 4 は、OUT_timeについて説明する図である。
- 図 3 5 は、Connection_Conditionのテーブルを示す図である。
- 図 3 6 A～図 3 6 D は、Connection_Conditionについて説明する図である。
- 図 3 7 は、BridgeSequenceInfoを説明する図である。
- 図 3 8 は、BridgeSequenceInfoのシンタクスを示す図である。
- 図 3 9 は、SubPlayItemについて説明する図である。
- 図 4 0 は、SubPlayItemのシンタクスを示す図である。
- 図 4 1 は、SubPath_typeのテーブルを示す図である。
- 図 4 2 は、PlayListMarkのシンタクスを示す図である。
- 図 4 3 は、Mark_typeのテーブルを示す図である。
- 図 4 4 は、Mark_time_stampを説明する図である。
- 図 4 5 は、zzzzz.clipのシンタクスを示す図である。
- 図 4 6 は、ClipInfoのシンタクスを示す図である。
- 図 4 7 は、Clip_stream_typeのテーブルを示す図である。
- 図 4 8 は、offset_SPNについて説明する図である。
- 図 4 9 は、offset_SPNについて説明する図である。
- 図 5 0 A 及び図 5 0 B は、S T C 区間について説明する図である。
- 図 5 1 は、STC_Infoについて説明する図である。
- 図 5 2 は、STC_Infoのシンタクスを示す図である。
- 図 5 3 は、ProgramInfoを説明する図である。
- 図 5 4 は、ProgramInfoのシンタクスを示す図である。
- 図 5 5 は、VideoCondInfoのシンタクスを示す図である。
- 図 5 6 は、Video_formatのテーブルを示す図である。
- 図 5 7 は、frame_rateのテーブルを示す図である。
- 図 5 8 は、display_aspect_ratioのテーブルを示す図である。
- 図 5 9 は、AudioCondInfoのシンタクスを示す図である。

- 図 6 0 は、audio_codingのテーブルを示す図である。
- 図 6 1 は、audio_component_typeのテーブルを示す図である。
- 図 6 2 は、sampling_frequencyのテーブルを示す図である。
- 図 6 3 は、CPIについて説明する図である。
- 図 6 4 は、CPIについて説明する図である。
- 図 6 5 は、CPIのシンタクスを示す図である。
- 図 6 6 は、CPI_typeのテーブルを示す図である。
- 図 6 7 は、ビデオEP_mapについて説明する図である。
- 図 6 8 は、EP_mapについて説明する図である。
- 図 6 9 は、EP_mapについて説明する図である。
- 図 7 0 は、EP_mapのシンタクスを示す図である。
- 図 7 1 は、EP_type valuesのテーブルを示す図である。
- 図 7 2 は、EP_map_for_one_stream_PIDのシンタクスを示す図である。
- 図 7 3 は、TU_mapについて説明する図である。
- 図 7 4 は、TU_mapのシンタクスを示す図である。
- 図 7 5 は、ClipMarkのシンタクスを示す図である。
- 図 7 6 は、mark_typeのテーブルを示す図である。
- 図 7 7 は、mark_type_stampのテーブルを示す図である。
- 図 7 8 は、menu.thmbとmark.thmbのシンタクスを示す図である。
- 図 7 9 は、Thumbnailのシンタクスを示す図である。
- 図 8 0 は、thumbnail_picture_formatのテーブルを示す図である。
- 図 8 1 A 及び図 8 1 B は、tn_blockについて説明する図である。
- 図 8 2 は、DVR MPEG 2 のトランスポートストリームの構造について説明する図である。
- 図 8 3 は、DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。
- 図 8 4 は、DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。
- 図 8 5 は、source packetのシンタクスを示す図である。

- 図 8 6 は、TP_extra_headerのシンタクスを示す図である。
- 図 8 7 は、copy permission indicatorのテーブルを示す図である。
- 図 8 8 は、シームレス接続について説明する図である。
- 図 8 9 は、シームレス接続について説明する図である。
- 図 9 0 は、シームレス接続について説明する図である
- 図 9 1 は、シームレス接続について説明する図である。
- 図 9 2 は、シームレス接続について説明する図である
- 図 9 3 は、オーディオのオーバーラップについて説明する図である。
- 図 9 4 は、BridgeSequenceを用いたシームレス接続について説明する図である。
- 図 9 5 は、BridgeSequenceを用いないシームレス接続について説明する図である。
- 図 9 6 は、DVR STDモデルを示す図である。
- 図 9 7 は、復号、表示のタイミングチャートを示す図である。
- 図 9 8 は、BridgeSequenceInfoの他のシンタクスを示す図である。
- 図 9 9 は、2つのPlayItemがシームレスに接続されときのBridge-Clipについて説明する図である。
- 図 1 0 0 は、ClipInformationファイルのシンタクスを示す図である。
- 図 1 0 1 は、ClipInformationファイルのClipInfoのシンタクスを示す図である。
- 図 1 0 2 は、ClipInformationファイルのSequenceInfoのシンタクスを示す図である。
- 図 1 0 3 A及び図 1 0 3 Bは、ClipAVストリームファイルのストリームデータを部分的に消去した場合のデータベースの変更について説明する図である。
- 図 1 0 4 は、RealPlayListの作成について説明するフローチャートである。
- 図 1 0 5 は、VirtualPlayListの作成について説明するフローチャートである。
- 図 1 0 6 は、ブリッジシーケンスの作成について説明するフローチャートである。
- 図 1 0 7 は、PlayListの再生について説明するフローチャートである。
- 図 1 0 8 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された情報処理装置及び方法、プログラム、並びに記録媒体について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置1は、アナログデータ、又は、デジタルデータを入力し、記録することができる構成となっている。

端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、AVエンコーダ15に出力される。解析部14は、入力されたビデオ信号からシーンチェンジ等の特徴点を抽出する。

AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、及びAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケッタイザ19に出力する。

多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC(誤り訂正)符号化部20、変調部21で所定の処理が施され、書込部22に出力される。書込部22は、制御部23から出

力される制御信号に基づいて、記録媒体 100 に AV ストリームファイルを書き込む（記録する）。

ディジタルインタフェース又はディジタルテレビジョンチューナから入力されるディジタルテレビジョン放送等のトランスポートストリームは、端子 13 に入力される。端子 13 に入力されたトランスポートストリームの記録方式には、2通りあり、これらは、トランスペアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子 24 から制御部 23 へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子 13 に入力されたトランスポートストリームは、多重化ストリーム解析部 18 と、ソースパケットタイザ 19 に出力される。これ以降の記録媒体 100 へ AV ストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子 13 に入力されたトランスポートストリームは、デマルチプレクサ 26 に入力される。デマルチプレクサ 26 は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム（V）、オーディオストリーム（A）、及びシステム情報（S）を抽出する。

デマルチプレクサ 26 により抽出されたストリーム（情報）の内、ビデオストリームは AV デコーダ 27 に、オーディオストリームとシステム情報はマルチプレクサ 16 に、それぞれ出力される。AV デコーダ 27 は、入力されたビデオストリームを復号し、その再生ビデオ信号を AV エンコーダ 15 に出力する。AV エンコーダ 15 は、入力ビデオ信号を符号化し、符号化ビデオストリーム（V）をマルチプレクサ 16 に出力する。

一方、デマルチプレクサ 26 から出力され、マルチプレクサ 16 に入力されたオーディオストリームとシステム情報、及び、AV エンコーダ 15 から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部 18 とソースパケットタイザ 19 にスイ

ッチ 17 を介して出力される。これ以後の記録媒体 100 へ A V ストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

本記録再生装置 1 は、A V ストリームのファイルを記録媒体 100 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部 23 により作成される。制御部 23 への入力情報は、解析部 14 からの動画像の特徴情報、多重化ストリーム解析部 18 からの A V ストリームの特徴情報、及び端子 24 から入力されるユーザからの指示情報である。

解析部 14 から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関係する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル (CM) の開始・終了点等の指定情報 (マーク) であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

多重化ストリーム解析部 18 からの A V ストリームの特徴情報は、記録される A V ストリームの符号化情報に関係する情報であり、例えば、A V ストリーム内の I ピクチャのアドレス情報、A V ストリームの符号化パラメータ、A V ストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関係する情報 (マーク) 等である。

端子 24 からのユーザの指示情報は、A V ストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報等である。

制御部 23 は、上記の入力情報に基づいて、A V ストリームのデータベース (Clip)、A V ストリームの再生区間 (PlayItem) をグループ化したもの (PlayList) のデータベース、記録媒体 100 の記録内容の管理情報 (info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、A V ストリームと同様にして、E C C 符号化部 20、変調部 21 で処理されて、書込部 22 へ入力される。書込部 22 は、制御部 23 から出力される制御信号に基づいて、記録媒体 100 へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生される場合、先ず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29、ECC復号部30の処理を経て、制御部23へ入力される。

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読出部28に指示する。読出部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、更にECC復号部30の処理を経て、ソースデパケッタ31出力される。

ソースデパケッタ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26に出力できるストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、及びAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読出部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制

御部 2 3 は、指定された時刻に最も近いタイムスタンプを持つ I ピクチャからのデータを読み出すように読出部 2 8 に指示する。

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部 2 3 は、A V ストリームのデータベース(Clip)に基づいて、A V ストリームの中の I ピクチャデータを順次連続して読み出すように読出部 2 8 に指示する。

読出部 2 8 は、指定されたランダムアクセスポイントから A V ストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体 1 0 0 に記録されている A V ストリームの編集をする場合を説明する。ユーザが、記録媒体 1 0 0 に記録されている A V ストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組 A という歌番組から歌手 A の部分を再生し、その後続けて、番組 B という歌番組の歌手 A の部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子 2 4 から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部 2 3 に入力される。制御部 2 3 は、A V ストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体 1 0 0 に記録されている A V ストリームの一部を消去したい場合、ユーザインタフェースとしての端子 2 4 から消去区間のイン点とアウト点の情報が制御部 2 3 に入力される。制御部 2 3 は、必要な A V ストリーム部分だけを参照するように PlayList のデータベースを変更する。また、A V ストリームの不必要なストリーム部分を消去するように、書込部 2 2 に指示する。

ユーザが、記録媒体 1 0 0 に記録されている A V ストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部 2 3 は、A V ストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリートの部分的な再エンコードと再多重化を行う。

まず、端子 2 4 から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部 2 3 へ入力される。制御部 2 3 は、読出部 2 8 にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示す

る。そして、読出部 28 は、記録媒体 100 からデータを読み出し、そのデータは、復調部 29、ECC 復号部 30、ソースデパケッタイザ 31 を経て、デマルチプレクサ 26 に出力される。

制御部 23 は、デマルチプレクサ 26 に入力されたデータを解析して、ビデオストリームの再エンコード方法 (picture_coding_type の変更、再エンコードする符号化ビット量の割り当て) と、再多重化方式を決定し、その方式を AV エンコーダ 15 とマルチプレクサ 16 に供給する。

次に、デマルチプレクサ 26 は、入力されたストリームをビデオストリーム (V)、オーディオストリーム (A)、及びシステム情報 (S) に分離する。ビデオストリームは、「AV デコーダ 27 に入力されるデータ」と「マルチプレクサ 16 に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これは AV デコーダ 27 で復号され、復号されたピクチャは AV エンコーダ 15 で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ 16 に入力される。

マルチプレクサ 16 は、制御部 23 から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC 符号化部 20、変調部 21 で処理されて、書込部 22 に入力される。書込部 22 は、制御部 23 から供給される制御信号に基づいて、記録媒体 100 に AV ストリームを記録する。

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図 2 は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AV ストリームの管理のために Playlist と Clip の 2 つのレイヤを持つ。Volume Information は、ディスク内の全ての Clip と Playlist の管理をする。ここでは、1 つの AV ストリームとその付属情報のペアを 1 つのオブジェクトとし、それを Clip という。AV ストリームファイルは Clip AV stream file といい、その付属情報は、Clip Information file という。

1つのClip AV stream fileは、MPEG 2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、これは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4Aは、Real PlayListのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4Bは、Real PlayListのディバイド(divide:分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番

組として登録（記録）し直したいといったようなときに行われる。この操作により、Clipの内容が変更される（Clip自体が分割される）ことはない。

図4 Cは、Real PlayListのコンバイン(combine:結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が1つにされる）ことはない。

図5 Aは、Real PlayList全体のデリート(delete:削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図5 Bは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図5 Cは、Real PlayListのミニマイズ(Minimize:最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayListにとって不必要なClipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

このようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか?」といったメッセージ等を表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次にVirtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アセンブル(Assemble)編集(IN-OUT編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual Playlistを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

図6Aに示したように、2つのReal Playlist1, 2と、それぞれのReal Playlistに対応するClip1, 2が存在している場合に、ユーザがReal Playlist1内の所定の区間(In1乃至Out1までの区間: PlayItem1)を再生区間として指示し、続けて再生する区間として、Real Playlist2内の所定の区間(In2乃至Out2までの区間: PlayItem2)を再生区間として指示したとき、図6Bに示すように、PlayItem1とPlayItem2から構成される1つのVirtual Playlistが作成される。

次に、Virtual Playlistの再編集(Re-editing)について説明する。再編集には、Virtual Playlistの中のイン点やアウト点の変更、Virtual Playlistへの新しいPlayItemの挿入(insert)や追加(append)、Virtual Playlistの中のPlayItemの削除等がある。また、Virtual Playlistそのものを削除することもできる。

図7は、Virtual Playlistへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual Playlistへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlaylistのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

Real PlaylistとVirtual Playlistで共通の操作として、図8に示すようなPlaylistの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlaylistの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of Playlist(図20等を参照して後述する)によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク(Mark)について説明する。マークは、Clip及びPlaylistの中の

ハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、A Vストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点等である。PlayListを再生するとき、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点等である。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

次に、サムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）等を利用して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットしたときに、そのディスクの内容を表す静止画を最初に表示する場合等に用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出さる必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画

から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル (Mark Thumbnail) という。したがって、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表すときに、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、Playlist、及びClipの関係について示した図である。メニューサムネイルファイルには、Playlist毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各Playlist毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用い

られる。本例では、2種類のCPIを用いる。1つは、EP_mapであり、もう1つは、TU_mapである。

EP_mapは、エントリポイント（EP）データのリストであり、それはエレメンタリーストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

本例では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

SESFは、MPEG-2トランスポートストリーム及びAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式の内のいずれかが用いられて記

録媒体 100 に記録される。まず、デジタル放送のストリームを S E S F ストリームにトランスコーディングする。この場合、記録されたストリームは、S E S F に準拠しなければならない。この場合、EP_map が作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_map が作成されて、ディスクに記録されなければならない。

例えば、入力ストリームが I S D B (日本のデジタル B S 放送の規格名称) 準拠の M P E G - 2 トランスポートストリームであり、それが H D T V ビデオストリームと M P E G A A C オーディオストリームを含むとする。H D T V ビデオストリームを S D T V ビデオストリームにトランスコーディングし、その S D T V ビデオストリームとオリジナルの A A C オーディオストリームを T S に再多重化する。S D T V ストリームと記録されるトランスポートストリームは、共に I S D B フォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体 100 に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する (入力トランスポートストリームを何も変更しないで記録する) 場合であり、そのときに EP_map が作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する (入力トランスポートストリームを何も変更しないで記録する) 場合であり、そのときに TU_map が作成されてディスクに記録される。

次に、ディレクトリとファイルについて説明する。以下、記録再生装置 1 を D V R (Digital Video Recording) と適宜記述する。図 14 はディスク上のディレクトリ構造の一例を示す図である。D V R のディスク上に必要なディレクトリは、図 14 に示したように、"DVR"ディレクトリを含む root ディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、及び"DATA"ディレクトリを含む"DVR"ディレクトリである。root ディレクトリの下に、これら以外

のディレクトリを作成されるようにしてもよいが、それらは、本例のアプリケーションフォーマットでは、無視されたとする。

”DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。”DVR”ディレクトリは、4個のディレクトリを含む。”PLAYLIST”ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

”CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。”M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。”DATA”ディレクトリは、デジタルTV放送等のデータ放送のファイルがストアされる。

”DVR”ディレクトリは、次に示すファイルをストアする。”info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されたとする。”menu.thmb”ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されたとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

”mark.thmb”ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されたとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

”PLAYLIST”ディレクトリは、2種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。”xxxxx.rpls”ファイルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、”xxxxx.rpls”である。ここで、”xxxxx”は、5個の0乃至9まで数字である。ファイル拡張子は、”rpls”で

なければならないとする。

"yyyyy.vpls"ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、5個の0乃至9まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file 又は Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。仮に、AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻を、PTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。

更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリーム

の同期再生のために、ただ1つのシステムタイムベースを共有するものである。記録再生装置1にとって、AVストリームのデコードに先だち、そのAVストリームの内容が分かることは有用である。例えば、ビデオやオーディオのエレメンタリーストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリーム等）等の情報である。

この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリーストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。

ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceと呼ぶ。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

”DATA”ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイル等である。

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、”info.dvr”ファイルについて説明する。図15は、”info.dvr”ファイルのシンタクスを示す図である。”info.dvr”ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、及びMakerPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明すると、TableOfPlayLists

_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、info.dvrのシンタクスに従って挿入される。N 1とN 2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

DVRVolume()は、ボリューム (ディスク) の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVR Volume()のシンタクスを説明すると、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

ResumeVolume()は、ボリュームの中で最後に再生したReal PlayList又はVirtual PlayListのファイル名を記憶している。但し、Real PlayList又はVirtual PlayListの再生をユーザが中断したときの再生位置は、PlayListMark()において定義されるresume-markにストアされる。

図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明すると、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlayList又はVirtual PlayListのファイル名を示す。

図16に示したDVRVolume()のシンタクスの中の、UIAppInfoVolume は、ボリュームについてのユーザインタフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティク

スを説明すると、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlaylistの一覧を表示させる。それぞれのPlaylistの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlaylist()の中に定義されるplayback_control_flagによって示される。

PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説

明する。TableOfPlayLists()は、Playlist(Real PlaylistとVirtual Playlist)のファイル名をストアする。ボリュームに記録されている全てのPlaylistファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlaylistのデフォルトの再生順序を示す。

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明すると、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、Playlist_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist_file_nameの10バイトの数字は、Playlistのファイル名を示す。

図21は、TableOfPlayLists()のシンタクスの他の例を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlaylist(後述)を含ませた構成とされている。このように、UIAppinfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでもよい。

所定のメーカーが、プライベートデータを挿入したいときに、既に他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデ

ータをMakersPrivateData()の中に追加するようにする。このように、本例においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明すると、version_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数は0からカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカープライベートデータのエン트리数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

maker_model_codeは、そのメーカープライベートデータを作成したDVRシステムのモデルナンバコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。start_mpd_block_numberは、そのメーカープライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカープライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。

ない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real PlayList)、又は、yyyyy.vpls (Virtual PlayList) のシンタクスを示す図である。xxxx.rplsとyyyyy.vplsは、同一のシンタクス構成を持つ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、PlayList()、PlayListMark()、及びMakerPrivateData()である。

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding_word (パディングワード) は、PlayListファイルのシンタクスに従って挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

ここで、既に、簡便に説明したが、PlayListについて更に説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip (後述) を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24を参照して更に説明すると、図24Aに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、何れかのReal PlayListを参照することにより、必ず視聴することが可能である。

図 2 4 C に示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

Real Playlistは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual Playlistは、PlayItemのリストを含み、Playlist()の中に示されるCPI_typeがEP_map typeであり、且つPlaylist_typeが0（ビデオとオーディオを含むPlaylist）である場合、Virtual Playlistは、1つのSubPlayItemを含むことができる。本例におけるPlaylist()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual Playlistが持つSubPlayItemの数は、0又は1でなければならない。

次に、Playlistについて説明する。図 2 5 は、Playlistのシンタクスを示す図である。図 2 5 に示したPlaylistのシンタクスを説明すると、version_numberは、このPlaylist()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlaylist()の最後まで Playlist()のバイト数を示す32ビットの符号なし整数である。Playlist_typeは、このPlaylistのタイプを示す8ビットのフィールドであり、その一例を図 2 6 に示す。

CPI_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlaylistによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、Playlistの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、Playlistの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームのバス（オーディオストリームバス）は、サブバス的一种である。

次に、図 2 5 に示したPlaylistのシンタクスのUIAppInfoPlaylistについて説明

する。UIAppInfoPlayListは、PlayListについてのユーザインタフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明すると、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

name_lengthは、8ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録されたときの日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker_codeは、

そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

write_protect_flagは、図28Aにテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、そのReal PlayListを”Minimize”する。

is_played_flagは、図28Bに示すように、フラグが1にセットされている場合、そのPlayListは、記録されてから1度は再生されたことを示し、0にセットされている場合、そのPlayListは、記録されてから1度も再生されたことがないことを示す。

archiveは、図28Cに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でre

f_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、及び、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

PlayListが2つ以上のPlayItemから構成されるとき、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一系列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持たないとき、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図29に示す。

図30は、PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持つとき、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time (図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time (図の中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time (図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time (図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を

指している。

図 3 1 に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AV ストリーム上の時間を指している。

PlayItemのシンタクスは、図 3 2 に示すようになる。図 3 2 に示したPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

STC_sequence_idは、8 ビットのフィールドであり、PlayItemが参照するSTC 連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この8 ビットフィールドは何も意味を持たず、0 にセットされる。IN_timeは、3 2 ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図 3 3 に示すように、PlayList()において定義されるCPI_typeによって異なる。

OUT_timeは、3 2 ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図 3 4 に示すように、PlayList()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図 3 5 に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2 ビットのフィールドである。図 3 6 A ~ 図 3 6 D は、図 3 5 に示したConnection_Conditionの各状態について説明する図である。

次に、BridgeSequenceInfoについて、図 3 7 を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Cl

ipと称される。更に現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明すると、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使

用は、PlayList()のCPI_typeがEP_map typeである場合だけに許される。本例においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、S T C不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をスト

アする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。 $Presentation_end_TS = PTS_out + AU_duration$ ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90 kHz単位の表示期間である。

次に、図23に示したxxxx.rplsとyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明すると、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、

それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル) は、図45に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、及びMakerPrivateData()である。AVストリーム(Clip AVストリーム又はBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の”zzzzz”が使用される。

図45に示したzzzzz.clpi (Clip information fileファイル) のシンタクスについて説明すると、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数は0からカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数は0からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData ()の先頭アドレ

スを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、zzzzz.clpiファイルのシンタクスに従って挿入される。N 1, N 2, N 3, N 4、及びN 5は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図4 6は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリーム又はBridge-Clip AVストリームファイル) の属性情報をストアする。

図4 6に示したClipInfoのシンタクスについて説明すると、version_numberは、このClipInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図4 7に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリーム又はBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

図4 8に示すように、AVストリームファイルのはじめの部分が編集によって消去されたとき、offset_SPNは、0以外の値をとってもよい。本例では、offset_SPNを参照する相対ソースパケット番号 (相対アドレス) が、しばしば、RSPN_xxx (xxxは変形する。例、RSPN_EP_start) の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN_xxx) は、次式で算出される。

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

図 4 8 に、 offset_SPN が、 4 である場合の例を示す。

TS_recording_rate は、 24 ビットの符号なし整数であり、この値は、DVR ドライブ（書込部 22）へ又は DVR ドライブ（読出部 28）からの AV ストリームの必要な入出力のビットレートを与える。record_time_and_date は、Clip に対応する AV ストリームが記録されたときの日時をストアする 56 ビットのフィールドであり、年／月／日／時／分／秒について、14 個の数字を 4 ビットの Binary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、"0x20011223010203" と符号化される。

duration は、Clip の総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した 24 ビットのフィールドである。このフィールドは、6 個の数字を 4 ビットの Binary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30 は、"0x014530" と符号化される。

time_controlled_flag のフラグは、AV ストリームファイルの記録モードを示す。この time_controlled_flag が 1 である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\begin{aligned} \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) - \alpha &\leq \text{size_clip}(t) \\ &\leq \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha \end{aligned}$$

ここで、TS_average_rate は、AV ストリームファイルのトランスポートストリートの平均ビットレートを bytes/second の単位で表したものである。

また、上式において、t は、秒単位で表される時間を示し、start_time は、AV ストリームファイルの最初のソースパケットが記録されたときの時刻であり、秒単位で表される。size_clip(t) は、時刻 t における AV ストリームファイルのサイズをバイト単位で表したものであり、例えば、start_time から時刻 t までに 10 個のソースパケットが記録された場合、size_clip(t) は 10 * 192 バイトである。 α は、TS_average_rate に依存する定数である。

time_controlled_flag が 0 にセットされている場合、記録モードは、記録の時間経過と AV ストリームのファイルサイズが比例するように制御していないことを

示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。先ずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

に基づいて算出される。

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1であるとき、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_service_ID_validのフラグが1である場合、service_IDのフィールドが有効であることを示す。

is_country_code_validのフラグが1であるとき、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration_descriptor (ISO/IEC 13818-1で定義されている) が持つformat_identifierの値を示す。original_network

k_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

servece_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservece_IDの値を示す。country_codeの24ビットのフィールドは、ISO 3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO 8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

format_identifier、original_network_ID、transport_stream_ID、servece_ID、country_code、及びstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI（サービスインフォメーション）の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceといい、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（但し、後述するように、Clipの最大時間長は制限されている）。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目（ $k \geq 0$ ）のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、（ $k + 1$ ）番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明すると、version_numberは、このSTC_Info()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0をセットしてもよい。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウント

される。そのAV streamファイルの中での絶対アドレスは、既に上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。

次に、図 4 5 に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図 5 3 を参照しながら説明すると、ここでは、Clipの中で次の特徴を持つ時間区間をprogram_sequenceと呼ぶ。まず、PCR_P I Dの値が変わらない。次に、ビデオエレメンタリーストリームの数が増減しない。また、それぞれのビデオストリームについてのP I Dの値とそのVideoCodingInfoによって定義される符号化情報が増減しない。更に、オーディオエレメンタリーストリームの数が増減しない。また、それぞれのオーディオストリームについてのP I Dの値とそのAudioCodingInfoによって定義される符号化情報が増減しない。

program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

図 5 4 は、ProgramInfoのシンタクスを示す図である。図 5 4 に示したProgramInfoのシンタクを説明すると、version_numberは、このProgramInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、I S O 6 4 6 に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0にセットされてもよい。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは1以上の値でなければならない。

number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが増減しない場合、number_of_program_sequencesは1にセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが

開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、A Vストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのA Vストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明すると、video_formatの8ビットフィールドは、図56に示すように、ProgramInf

o()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明すると、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_map又は1つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明すると、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタ

イブを表す。

次に、図 6 5 に示した C P I のシンタクス内の EP_map について説明する。EP_map には、2 つのタイプがあり、それはビデオストリーム用の EP_map とオーディオストリーム用の EP_map である。EP_map 中の EP_map_type が、EP_map のタイプを区別する。Clip が 1 つ以上のビデオストリームを含む場合、ビデオストリーム用の EP_map が使用されなければならない。Clip がビデオストリームを含まず、1 つ以上のオーディオストリームを含む場合、オーディオストリーム用の EP_map が使用されなければならない。

ビデオストリーム用の EP_map について図 6 7 を参照して説明する。ビデオストリーム用の EP_map は、stream_PID、PTS_EP_start、及び、RSPN_EP_start というデータを持つ。stream_PID は、ビデオストリームを伝送するトランスポートパケットの P I D を示す。PTS_EP_start は、ビデオストリームのシーケンスヘッダから始めるアクセスユニットの P T S を示す。RSPN_EP_start は、A V ストリームの中で PTS_EP_start により参照されるアクセスユニットの第 1 バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID() と呼ばれるサブテーブルは、同じ P I D を持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clip の中に複数のビデオストリームが存在する場合、EP_map は複数の EP_map_for_one_stream_PID() を含んでもよい。

オーディオストリーム用の EP_map は、stream_PID、PTS_EP_start、及び RSPN_EP_start というデータを持つ。stream_PID は、オーディオストリームを伝送するトランスポートパケットの P I D を示す。PTS_EP_start は、オーディオストリームのアクセスユニットの P T S を示す。RSPN_EP_start は、A V ストリームの中で PTS_EP_start で参照されるアクセスユニットの第 1 バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID() と呼ばれるサブテーブルは、同じ P I D を持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clip の中に複数のオーディオストリームが存在する場合、EP_map は複数の EP_map_for_one_stream_PID() を含んでもよい。

EP_mapとSTC_Infoの関係を説明すると、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図68を参照）。・EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明すると、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数に持つfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリestream（ビデオ又はオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0('video')に等しい場合、そのエレメンタリestreamはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリestreamはオーディオストリームでなければならない。

num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

padding_wordは、EP_map()のシンタクスに従って挿入されなければならない。
XとYは、0 又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取ってもよい。

図 7 2 は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図 7 2 に示したEP_map_for_one_stream_PIDのシンタクスを説明すると、PTS_EP_startの 32 ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの 3.3 ビット精度のPTSの上位 32 ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの 3.3 ビット精度のPTSの上位 32 ビットを持つ。

RSPN_EP_startの 32 ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースボケットの相対アドレスを示す。又は、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースボケットの相対アドレスを示す。

RSPN_EP_startは、ソースバケット番号を単位とする大きさであり、AVストリームファイルの最初のソースバケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図 7 3 を参照して説明する。TU_mapは、ソースバケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、0

ffset_timeから一定の単位に分割される。その単位を、time_unitという。

A Vストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのA Vストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startという。TU_map_time_axis上において、k (k >=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$TU_start_time(k) = offset_time + k * time_unit_size$$

TU_start_time(k)は、4 5 K H z の精度を持つ。

図 7 5 は、TU_mapのシンタクスを示す図である。図 7 5 に示したTU_mapのシンタクスを説明すると、offset_timeの3 2 bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、2 7 M H z 精度のアライバルタイムクロックから導き出される4 5 kHzクロックを単位とする大きさである。A Vストリームが新しいClipとして記録される場合、offset_timeは0にセットされなければならない。

time_unit_sizeの3 2 ビットフィールドは、time_unitの大きさを与えるものであり、それは2 7 M H z 精度のアライバルタイムクロックから導き出される4 5 K H z クロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size <= 45000) にすることがよい。number_of_time_unit_entriesの3 2 ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリ数を示す。

RSPN_time_unit_startの3 2 ビットフィールドは、A Vストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$SPN_xxx = RSPN_xxx - offset_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k + 1) 番目のtime_unitの中にソースパケット

が何もない場合、 $(k + 1)$ 番目のRSPN_time_unit_startは、 k 番目のRSPN_time_unit_startと等しくなければならない。

図 4 5 に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置 1）によってセットされるものであり、ユーザによってセットされるものではない。

図 7 5 は、ClipMarkのシンタクスを示す図である。図 7 5 に示したClipMarkのシンタクスを説明すると、version_numberは、このClipMark()のバージョンナンバーを示す 4 個のキャラクター文字である。version_numberは、ISO 646 に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す 3 2 ビットの符号なし整数である。number_of_clip_marksは、ClipMarkの中にストアされているマークの個数を示す 1 6 ビットの符号なし整数。number_of_clip_marks は、0 であってもよい。mark_typeは、マークのタイプを示す 8 ビットのフィールドであり、図 7 6 に示すテーブルに従って符号化される。

mark_time_stampは、3 2 ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図 7 7 に示すように、Playlist()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この 8 ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この 8 ビットのフィールドは何も意味を持たず、0 にセットされる。character_setの 8 ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図 1 9 に示される値に対応する。

name_lengthの 8 ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、及び、それぞれのPlayListを代表する画像をストアする。全てのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

図78は、menu.thmbとmark.thmbのシンタクスを示す図であり、図79は、図78に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図79に示したThumbnailのシンタクスについて説明すると、version_numberは、このThumbnail()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersP

privateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、及びClipMark()の中のref_thumbnail_indexによって参照される。

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図80に示すような値をとる。表中のDCFとPNGは”menu.thmb”内でのみ許される。マークサムネイルは、値”0x00”(MPEG-2 Video I-picture)をとらなければならない。

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tn_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全

てのtn_blockは、同じサイズ（固定長）であり、その大きさはtn_block_sizeによって定義される。

図 8 1 A 及び図 8 1 B は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図 8 1 A 及び図 8 1 B のように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、“M2TS”ディレクトリ（図 1 4）にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2 トランスポートストリームファイルの構造でなければならない。

まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2 トランスポートストリームの構造は、図 8 2 に示すようになっている。AVストリームファイルは、DVR MPEG 2 トランスポートストリームの構造を持つ。DVR MPEG 2 トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6 1 4 4 バイト（2048*3 バイト）である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、1 9 2 バイト長である。1つのソースパケットは、TP_extra_headerとトランスポートパケットからなる。TP_extra_headerは、4 バイト長であり、またトランスポートパケットは、1 8 8 バイト長である。

1つのAligned unitは、3 2 個のソースパケットからなる。DVR MPEG 2 トランスポートストリームの中の最後のAligned unitも、また3 2 個のソースパケットからなる。よって、DVR MPEG 2 トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が3 2 の倍数でないとき、ヌルパケット（PID=0x1FFFのトランスポートパケット）を持ったソースパケットを最後のAligned un

itに使用しなければならない。ファイルシステムは、DVR MPEG 2 トラストポートストリームに余分な情報を付加してはならない。

図 8 3 に、DVR MPEG-2 トラストポートストリームのレコーダモデルを示す。図 8 3 に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2 トラストポートストリームは、このモデルに従う。

MPEG-2 トラストポートストリームの入力タイミングについて説明する。入力MPEG 2 トラストポートストリームは、フルトラストポートストリーム又はパーシャルトラストポートストリームである。入力されるMPEG 2 トラストポートストリームは、ISO/IEC 13818-1 又は ISO/IEC 13818-9 に従っていなければならない。MPEG 2 トラストポートストリームの i 番目のバイトは、T-STD (ISO/IEC 13818-1 で規定される Transport stream system target decoder) とソースパケットタイザへ、時刻 $t(i)$ に同時に入力される。Rpkは、トラストポートパケットの入力レートの瞬時的な最大値である。

27MHz PLL 5 2 は、27MHz クロックの周波数を発生する。27MHz クロックの周波数は、MPEG-2 トラストポートストリームのPCR (Program Clock Reference) の値にロックされる。arrival time clock counter 5 3 は、27MHz の周波数のパルスをカウントするバイナリーカウンタである。Arrival_time_clock(i) は、時刻 $t(i)$ における Arrival time clock counter のカウント値である。

source packetizer 5 4 は、全てのトラストポートパケットにTP_extra_header を付加し、ソースパケットを作る。Arrival_time_stampは、トラストポートパケットの第 1 バイト目がT-STDとソースパケットタイザの両方へ到着する時刻を表す。Arrival_time_stamp(k) は、次式で示されるように Arrival_time_clock(k) のサンプル値であり、ここで、 k はトラストポートパケットの第 1 バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

2つの連続して入力されるトラストポートパケットの時間間隔が、230/270000 00秒 (約40秒) 以上になる場合、その2つのトラストポートパケットのarrival_

time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコードは、そのようになる場合に備えてある。

smoothing buffer 5 5 は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファは、オーバーフローしてはならない。Rmaxは、スムージングバッファが空でないときのスムージングバッファからのソースパケットの出力ビットレートである。スムージングバッファが空であるとき、スムージングバッファからの出力ビットレートは0である。

次に、DVR MPEG-2 トランスポートストリームのレコードモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$Rmax = TS_recording_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptor等において定義される値を参照してもよい。

smoothing buffer sizeは、入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファの大きさは0である。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptor等において定義される値を参照してもよい。

記録機（レコード）及び再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytes である。

次に、DVR MPEG-2 トランスポートストリームのプレーヤモデルについて説明する。図84は、DVR MPEG-2 トランスポートストリームのプレー

ヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

27MHz X-tal 61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 ± 30 ppm (27000000 \pm 810 Hz)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

smoothing buffer 64において、Rmaxは、スムージングバッファがフルでないときのスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルであるとき、スムージングバッファへの入力ビットレートは0である。

MPEG-2トランスポートストリームの出力タイミングを説明すると、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しいとき、そのソースパケットのトランスポートパケットは、スムージングバッファから引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファは、アンダーフローしてはならない。

DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

図85は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図85に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図86に示す。図86に示したTP_Extra_headerのシンタクスについて説明すると、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、又はcopy prohibitedとすることができる。図87は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

copy_permission_indicatorは、全てのトランスポートパケットに付加される。IEEE 1394デジタルインタフェースを使用して入力トランスポートスト

リームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けてもよい。IEEE 1394 デジタルインタフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けてもよい。

arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

において、arrival_time_stampによって指定される値を持つ整数値である。

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS (Presentation Time Stamp) が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない。且つ後述するDVR-STDに従わなければならない。

本例においては、編集におけるPlayItem間のビデオとオーディオのシームレス

接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図88は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図88においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図88に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータからなる。

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図88においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースバケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースバケットから、RSPN_arrival_time_discontinuityで参照されるソースバケットの直前のソースバケットまでのストリームデータである。

また、図88におけるTS2は、Clip2 (Clip AVストリーム) の影を付けら

れたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータからなる。TS 2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースバケットから、Bridge-Clipの最後のソースバケットまでのストリームデータである。TS 2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースバケットから、現在のPlayItemのOUT_time（図88においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図89は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図89におけるTS 1は、Clip1 (Clip AVストリーム)の影を付けられたストリームデータからなる。TS 1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図89においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1の最後のソースバケットまでのデータである。また、図89におけるTS 2は、Clip2 (Clip AVストリーム)の影を付けられたストリームデータからなる。

TS 2のClip2の影を付けられたストリームデータは、Clip2の最初のソースバケットから始まり、現在のPlayItemのOUT_time（図89においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図88と図89において、TS 1とTS 2は、ソースバケットの連続したストリームである。次に、TS 1とTS 2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS 1とTS 2の中に含まれるプログラムの数は、1でなければならない。TS 1とTS 2の中に含まれるビデオストリームの数は、1でなければならない。TS 1とTS 2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS 1と

T S 2の中に含まれるオーディオストリームの数は、等しくなければならない。
T S 1及び／又はT S 2の中に、上記以外のエレメンタリーストリーム又はサブ
イベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図90は、ピクチャの表示
順序で示すシームレス接続の例を示す図である。接続点においてビデオストリー
ムをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN
_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近の
Clipの部分的なストリームを再エンコードするプロセスにより、除去されなけれ
ばならない。

図90に示したような場合において、BridgeSequenceを使用してシームレス接
続を実現する例を、図91に示す。RSPN_arrival_time_discontinuityより前のB
ridge-Clipのビデオストリームは、図90のClip1のOUT_time1に対応するピクチャ
までの符号化ビデオストリームからなる。そして、そのビデオストリームは先
行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従っ
たエレメンタリーストリームとなるように再エンコードされている。

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオス
トリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオス
トリームからなる。そして、そのビデオストリームは、正しくデコード開始する
ことができ、これに続くClip2のビデオストリームに接続され、1つの連続でM
PEG2規格に従ったエレメンタリーストリームとなるように再エンコードされ
ている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードし
なければならず、それ以外のピクチャはオリジナルのClipからコピーすることが
できる。

図90に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実
現する例を図92に示す。Clip1のビデオストリームは、図90のOUT_time1に対
応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続で
MPEG2規格に従ったエレメンタリーストリームとなるように再エンコードさ
れている。同様にして、Clip2のビデオストリームは、図90のClip2のIN_time2
に対応するピクチャ以後の符号化ビデオストリームから成り、それは、1つの連

続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明すると、まず、TS 1とTS 2のビデオストリームのフレームレートは、等しくなければならない。TS 1のビデオストリームは、sequence_end_codeで終端しなければならない。TS 2のビデオストリームは、Sequence Header、GOP Header、そして、I-ピクチャで開始しなければならない。TS 2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあってはならない。接続点において、トップボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、“top_field_first”及び“repeat_first_field”フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明すると、TS 1とTS 2のオーディオのサンプリング周波数は、同じでなければならない。TS 1とTS 2のオーディオの符号化方法（例、MPEG 1レイヤ2、AC-3、SESF L PCM、AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明すると、TS 1のオーディオストリームの最後のオーディオフレームは、TS 1の最後の表示ピクチャの表示終了ときに等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS 2のオーディオストリームの最初のオーディオフレームは、TS 2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図93に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあってもよい。TS 2のエレメンタリーストリームを伝送する最初のバケットは、

ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明すると、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。

PlayItemの制限について説明すると、先行するPlayItemのOUT_time (図88、図89において示されるOUT_time1) は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (図88、図89において示されるIN_time2) は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図94を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているよう

に、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip A Vストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip A Vストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図95を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip A Vストリームファイル) の最後の部分とClip2 (Clip A Vストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1 (Clip A Vストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip A Vストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

所定のビットレートを持つデジタルA V信号が、ディスク上に断片化して記録されている場合、記録されたデジタルA V信号を所定のビットレートで記録媒体100から読み出せることを保証するためには、1つの連続記録領域の大きさが次の条件を満たさなければならない。

$$S \cdot 8 / (S \cdot 8 / R_{ud} + T_s) \geq R_{max}$$

ここで、

S : 1つの連続記録領域の最小の大きさ [Byte]

Ts: 1つの記録領域から次の記録領域へのフルストロークのアクセス時間[second]

Rud: 記録メディアからの読み出しビットレート [bit/second]

Rmax: A Vストリームのビットレート [bit/second]

すなわち、ディスク上で、Sバイト以上にA Vストリームのデータが連続して記録されるようにデータを配置しなければならない。

上記のハーフフラグメントの大きさが、Sバイト以上となるようにデータを配置しなければならない。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG 2 トラストストリーム生成及び検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルでもある。

DVR-STDモデルを図96に示す。図96に示したモデルには、DVR MPEG-2 トラストストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBSys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On及びPn(k)の表記方法は、ISO/IEC 13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデックス番号である。TBnは、エレメンタリーストリームnのトラストバッファである。

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBSysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBSysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2 トラストストリームを再生している間は、トラストパケ

ットをTB1, TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図88に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図97は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移るときのトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図97においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図97においてATC1で示される）と同じでない。

また、TS2のシステムタイムベースの時間軸（図97においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図97においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBn又はTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_

recording_rateの値である。TS 1の最後のバイトがバッファへ入力する時刻は、時刻T2である。したがって、時刻T1からT2までの区間では、ソースパケットのarrival_time_stampは無視される。

N1をTS 1の最後のビデオパケットに続くTS 1のトランスポートパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS 1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$\Delta T1 = T2 - T1 = N1 / TS_recording_rate(TS 1)$$

時刻T1乃至T2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS 1)の値に変化する。このルール以外のバッファリング動作は、TS STDと同じである。

T2の時刻において、arrival time clock counterは、TS 2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TBn又はTBsysのバッファへの入力タイミングは、TS 2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、TS STDにおいて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについて説明すると、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるよう、TS STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明すると、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS 1のシステムタイムベースの時間軸（図97ではSTC1と図示されている）とし、STC2は、TS 2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS 2の最初のPCRがTS STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、TS 1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS2startは、TS 2の最初のビデオプレゼンテーションユニットに対応す

る $STC2$ 上の PTS であり、 T_{pp} は、 $TS1$ の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセット STC_delta は、次式により算出される。

$$STC_delta = PTS1end + T_{pp} - PTS2start$$

オーディオのプレゼンテーションのタイミングについて説明すると、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である（図97に図示されている“audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

DVR-STDのシステムタイムクロックについて説明すると、時刻 $T5$ において、 $TS1$ の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻 $T2$ から $T5$ の間にオーバーラップしていてもよい。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値 ($STC1$) と新しいタイムベースの値 ($STC2$) の間で切り換える。 $STC2$ の値は、次式により算出される。

$$STC2 = STC1 - STC_delta$$

バッファリングの連続性について説明する。 $STC11video_end$ は、 $TS1$ の最後のビデオパケットの最後のバイトがDVR-STDの $TB1$ へ到着するときのシステムタイムベース $STC1$ 上の STC の値である。 $STC22video_start$ は、 $TS2$ の最初のビデオパケットの最初のバイトがDVR-STDの $TB1$ へ到着するときのシステムタイムベース $STC2$ 上の STC の値である。 $STC21video_end$ は、 $STC11video_end$ の値をシステムタイムベース $STC2$ 上の値に換算した値である。 $STC21video_end$ は、次式により算出される。

$$STC21video_end = STC11video_end - STC_delta$$

DVR-STDに従うために、次の2つの条件を満たすことが要求される。まず、 $TS2$ の最初のビデオパケットの $TB1$ への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC22video_start > STC21video_end + \Delta T1$$

この不等式が満たされるように、Clip 1 及び、又は、Clip 2 の部分的なストリームを再エンコード及び、又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC 1 と STC 2 を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS 1 からのビデオパケットの入力とそれに続く TS 2 からのビデオパケットの入力は、ビデオバッファをオーバーフロー及びアンダーフローさせてはならない。

図 9 8 は、BridgeSequenceInfo() のシンタックスの別例を示す図である。図 3 8 の BridgeSequenceInfo() との違いは、Bridge_Clip_Information_file_name しか含まれないことである。

図 9 9 は、図 9 8 の BridgeSequenceInfo() のシンタックスを使用する場合、2 つの PlayItem が、シームレスに接続されるとき Bridge-Clip について説明する図である。RSPN_exit_from_previous_Clip は、先行する PlayItem が参照する Clip AV stream 上のソースパケットのソースパケット番号であり、このソースパケットに続いて Bridge-Clip AV stream ファイルの最初のソースパケットが接続される。

RSPN_enter_to_current_Clip は、現在の PlayItem が参照する Clip AV stream 上のソースパケットの番号であり、このソースパケットの前に Bridge-Clip AV stream ファイルの最後のソースパケットが接続される。図 9 9 に示す Bridge-Clip AV ストリームファイルにおいて、SPN_ATC_start は、Bridge-Clip AV ストリームファイルの中で新しいアライバルタイムベースの時間軸が開始するソースパケットのソースパケット番号を示す。

Bridge-Clip AV ストリームファイルは 1 個のアライバルタイムベースの不連続点を持つ。図中で 2 番目の SPN_ATC_start は、図 3 7 の RSPN_arrival_time_discontinuity と同じ意味を持つ。

図 9 8 の BridgeSequenceInfo() のシンタックスを使用する場合、RSPN_exit_from_previous_Clip と RSPN_enter_to_current_Clip は、Bridge-Clip AV ストリームファイルに対応する Clip Information ファイルの中にストアされる。また、SPN_ATC_start もまた Clip Information ファイルの中にストアされる。

図100は、BridgeSequenceInfoが、図98のシンタクスの場合のClip Informationファイルのシンタクスを示す図である。SequenceInfo_start_addressは、Clip Informationファイルの先頭のバイトからの相対バイト数を単位として、SequenceInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。

図101は、図100のClip InformationファイルのClipInfo()のシンタクスを示す図である。Clip_stream_typeは、そのClipのAVストリームファイルがClip AVストリームファイルであるか、それともBridge-Clip AVストリームファイルであることを示す。Clip_stream_typeがBridge-Clip AVストリームファイルを示す場合、次のシンタクスフィールドが続く。

previous_Clip_Information_file_nameは、そのBridge-Clip AVストリームファイルの前に接続されるClipのClip Informationファイル名を示す。RSPN_exit_from_previous_Clipは、previous_Clip_Information_file_nameで示されるClip AVストリームファイル上のソースパケットのソースパケット番号であり、そのソースパケットに続いてBridge-Clip AVストリームファイルの最初のソースパケットが接続される。そのソースパケット番号は、Clip AVストリームファイルの最初のソースパケットから0を初期値としてカウントされる値である。

current_Clip_Information_file_nameは、そのBridge-Clip AVストリームファイルの後ろに接続されるClipのClip Informationファイル名を示す。RSPN_enter_to_current_Clipは、current_Clip_Information_file_nameで示されるClip AVストリームファイル上のソースパケットのソースパケット番号であり、そのソースパケットの前にBridge-Clip AVストリームファイルの最後のソースパケットが接続される。そのソースパケット番号は、Clip AVストリームファイルの最初のソースパケットから0を初期値としてカウントされる値である。

図102は、図100のClip InformationファイルのSequenceInfo()のシンタクスを示す。num_of_ATC_sequencesは、AVストリームファイルの中にあるATC-sequenceの数を示す。ATC-sequenceは、アライバルタイムベースの不連続点を含まないソースパケット列である。Bridge-Clipの場合、この値は2である。

SPN_ATC_start[atc_id] は、AVストリームファイル上でatc_idによって指されるアライバルタイムベースが開始するアドレスを示す。SPN_ATC_start[atc_i

d]は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから0を初期値としてカウントされる。

図103A及び図103Bは、Bridge-Sequenceによって参照されるClip AVストリームファイルのストリームデータを部分的に消去した場合のデータベースの変更について説明する図である。図103Aの”Before Editing”で示すように、Clip1とClip2がBridge-Clipで接続されていて、 $RSPN_exit_from_previous_Clip=X$, $RSPN_exit_from_previous_Clip=Y$ であるとする。

このとき、Clip1の斜線で示すZ1個のソースパケットのストリームデータ部分、及びClip2の斜線で示すZ2個のソースパケットのストリームデータ部分を消去するとする。その結果、図103Bの”After Editing”で示すように、 $RSPN_exit_from_previous_Clip=X-Z1$, $RSPN_exit_from_previous_Clip=Y-Z2$ に値が変更される。

BridgeSequenceに関係のあるデータベースのシンタクスを図98と図101のように変更することにより、AVストリーム中のデータアドレスを示すところのソースパケット番号についての情報（データベースのシンタクス中で、RSPNで始まるフィールド）がPlayListのレイヤからなくなり、ソースパケット番号の情報は全てClipのレイヤで記述されることになる。

これにより、AVストリーム中のデータアドレスの値に変更が必要になった場合（例えばAVストリームファイルのデータを部分的に消去したときにこれが必要になる）、Clipインフォメーションファイルだけをデータ管理すればよいので、データベースの管理が容易になるメリットがある。

図104は、Real PlayListの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS10において、制御部23はClip AVストリームを記録する。ステップS11において、制御部23は、上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlayList()を作成する。Clipの中にSTC不連続点があり、PlayList()が2つ以上のPlayItemからなる場合、PlayItem間のconnection_conditionもまた決定される。

ステップS12において、制御部23は、UIAppInfoPlayList()を作成する。ステップS13において、制御部23は、PlayListMarkを作成する。ステップ14

において、制御部 23 は、MakersPrivateDataを作成する。ステップ S 15 において、制御部 23 は、Real PlayList ファイルを記録する。このようにして、新規に Clip AV ストリームを記録する毎に、1 つの Real PlayList ファイルが作られる。

図 105 は、ブリッジシーケンスを持つ Virtual PlayList の作成について説明するフローチャートである。ステップ S 20 において、ユーザインタフェースを通して、ディスクに記録されている 1 つの Real PlayList の再生が指定される。そして、その Real PlayList の再生範囲の中から、ユーザインタフェースを通して、I N 点と O U T 点で示される再生区間が指定される。

ステップ S 21 において、制御部 23 は、ユーザによる再生範囲の指定操作が全て終了したか否かを判断し、終了していると判断した場合、ステップ S 22 に進み、終了していないと判断した場合、ステップ S 20 に戻り、それ以降の処理が繰り返される。

ステップ S 22 において、連続して再生される 2 つの PlayItem 間の接続状態 (connection_condition) を、ユーザがユーザインタフェースを通して決定するか、又は制御部 23 が決定する。ステップ S 23 において、制御部 23 は、シームレス接続される PlayItem のためのブリッジシーケンスを作成する。ステップ S 24 において、制御部 23 は、Virtual PlayList ファイルを作成し、記録する。

図 106 は、ステップ S 23 における詳細な処理を説明するフローチャートである。ステップ S 31 において、制御部 23 は、時間的に前側に表示される PlayItem の O U T 点側の AV ストリームの再エンコード及び再多重化を行う。ステップ S 32 において、制御部 23 は、上記 PlayItem に続いて表示される PlayItem の I N 点側の AV ストリームの再エンコード及び再多重化を行う。

ステップ S 33 において、制御部 23 は、データの連続供給のためのデータアロケーション条件を満たすように、RSPN_exit_from_previous_Clip の値を決定する。すなわち、RSPN_exit_from_previous_Clip 以前の Clip AV ストリームファイルのストリーム部分が、記録媒体上で前述のハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clip が選択されなければならない (図 91、図 94 を参照)。

ステップ S 34 において、制御部 23 は、データの連続供給のためのデータア

ロケーション条件を満たすように、RSPN_enter_to_current_Clipの値を決定する。すなわち、RSPN_enter_to_current_Clip以後のClip A Vストリームファイルのストリーム部分が、記録媒体100上で前述のハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない（図91、図94を参照）。

ステップS35において、制御部23は、データの連続供給のためのデータアロケーション条件を満たすように、Bridge-Clip A Vストリームファイルを作成する。すなわち、ステップS31とステップS32の処理で作成されたデータの量が、前述のハーフフラグメント以上のサイズ未満である場合、オリジナルのClipからデータがコピーされてBridge-Clipが作成される（図91、図94を参照）。

ステップS33，S34，S35の各処理は、時系列に説明しているが、これらの処理はお互いが関係するので、順不同もしくは同時に処理が行われてもよい。

ステップS36において、制御部23は、ブリッジシーケンスのデータベースを作成する。ステップS37において、制御部23は、Bridge-Clip A Vストリームファイル及びそのClipインフォメーションファイルを記録する。このようにして、ディスクに記録されているReal Playlistの再生範囲の中から、ユーザにより1つ以上のPlayItemが選択され、2つのPlayItem間がシームレス接続できるためのブリッジシーケンスが作成され、1つ以上のPlayItemがグループ化されたものを、1つのVirtual Playlistファイルとして記録される。

図107は、Playlistの再生について説明するフローチャートである。ステップS41において、制御部23は、Info.dvr, Clip Information file, Playlist file及びサムネイルファイルの情報を取得し、ディスクに記録されているPlaylistの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS42において、制御部23は、それぞれのPlaylistのUIAppInfoPlaylist()に基づいて、Playlistを説明する情報をGUI画面に提示する。ステップS43において、ユーザインタフェースを通して、GUI画面上からユーザが1つのPlaylistの再生を指示する。ステップS44において、制御部23は、現在のPlayItemのSTC-sequence-idとIN_timeのPTSから、IN_timeより時間的に前で

最も近いエントリポイントのあるソースパケット番号を取得する。

ステップS 4 5において、制御部2 3は、上記エントリポイントのあるソースパケット番号からAVストリームのデータを読み出し、デコーダへ供給する。ステップS 4 6において、現在のPlayItemの時間的に前のPlayItemがあった場合、制御部2 3は、前のPlayItemと現在のPlayItemとの表示の接続処理をconnection_conditionに従って行う。PlayItemがシームレス接続される場合、DVR-STDのデコード方法に基づいてAVストリームをデコードする。

ステップS 4 7において、制御部2 3は、AVデコーダ2 7にIN_timeのPTSのピクチャから表示を開始するように指示する。ステップS 4 8において、制御部2 3は、AVデコーダ2 7にAVストリームのデコードを続けるように指示する。ステップS 4 9において、制御部2 3は、現在表示の画像が、OUT_timeのPTSの画像か否かを判断し、OUT_timeのPTSの画像ではないと判断された場合、ステップS 5 0に進み、画像が表示された後、ステップS 4 8に戻り、それ以降の処理が繰り返される。

一方、ステップS 4 9において、現在表示の画像が、OUT_timeのPTSの画像であると判断された場合、ステップS 5 1へ進む。ステップS 5 1において、制御部2 3は、現在のPlayItemがPlayListの中で最後のPlayItemか否かを判断し、最後のPlayItemではないと判断された場合、ステップS 4 4に戻り、それ以降の処理が繰り返され、最後のPlayItemであると判断された場合、PlayListの再生を終了する。

このようにして、ユーザにより再生指示された1つのPlayListファイルの再生が行われる。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェア

に組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータ等に、記録媒体からインストールされる。

図108は、汎用のパーソナルコンピュータの内部構成例を示す図である。パーソナルコンピュータのCPU (Central Processing Unit) 201は、ROM (Read Only Memory) 202に記憶されているプログラムに従って各種の処理を実行する。RAM (Random Access Memory) 203には、CPU 201が各種の処理を実行する上において必要なデータやプログラム等が適宜記憶される。入出力インタフェース205は、キーボードやマウスから構成される入力部206が接続され、入力部206に入力された信号をCPU 201に出力する。また、入出力インタフェース205には、ディスプレイやスピーカ等から構成される出力部207も接続されている。

更に、入出力インタフェース205には、ハードディスク等から構成される記憶部208、及び、インターネット等のネットワークを介して他の装置とデータの授受を行う通信部209も接続されている。ドライブ210は、磁気ディスク221、光ディスク222、光磁気ディスク223、半導体メモリ224等の記録媒体からデータを読み出したり、データを書き込んだりするときに用いられる。

この記録媒体は、図108に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221 (フロッピーディスクを含む)、光ディスク222 (CD-ROM (Compact Disk-Read Only Memory), DVD (Digital Versatile Disk) を含む)、光磁気ディスク223 (MD (Mini-Disk) を含む)、若しくは半導体メモリ224等よりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM 202や記憶部208が含まれるハードディスク等で構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上の如く、本発明の情報処理装置及び方法、並びにプログラムによれば、第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、第1のAVストリームの所定の部分と第2のAVストリームの所定の部分から構成され、第1のAVストリームから第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成するようにしたので、別々に記録されたAVストリームの連続性を保つように再生できる。

また、本発明に係る情報処理装置及び方法、並びにプログラムによれば、第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームを記録媒体から読み出し、第3のAVストリームに関連する情報として、第1のAVストリームから第3のAVストリームに再生が切り換わるタイミングにおける第1のAVストリームのソースパケットのアドレスの情報と、第3のAVストリームから第2のAVストリームに再生が切り換わるタイミングにおける第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を記録媒体から読み出し、読み出された第3のAVストリームに関連する情報に基づいて第1のAVストリームから第3のAVストリームへ再生を切り換え、第3のAVストリームから第2のAVストリームへ再生を切り換えて再生するようにしたので、別々に記録されたAVストリームの連続性を保つように再生できる。

請求の範囲

1. 第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、前記第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成手段と、

前記生成手段により生成された前記第3のAVストリームと前記アドレス情報を記録する記録手段とを含む情報処理装置。

2. 前記生成手段により生成された前記アドレス情報に含まれる前記第1のAVストリームのソースパケットのアライバルタイムスタンプと、前記第3のAVストリームの最初に位置するソースパケットのアライバルタイムスタンプは連続しており、且つ、前記生成手段により生成された前記アドレス情報に含まれる前記第2のAVストリームのソースパケットのアライバルタイムスタンプと、前記第3のAVストリームの最後に位置するソースパケットのアライバルタイムスタンプは連続している請求の範囲第1項記載の情報処理装置。

3. 前記第3のAVストリーム内のソースパケットのアライバルタイムスタンプには、ただ1つの不連続点が存在する請求の範囲第2項記載の情報処理装置。

4. 前記生成手段により生成された前記アドレス情報に含まれる前記第1のAVストリームのソースパケットのアドレスの情報で示されるソースパケット以前のAVストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、前記アドレスは決定される請求の範囲第2項記載の情報処理装置。

５． 前記生成手段により生成された前記アドレス情報に含まれる前記第２のＡＶストリームのソースバケットのアドレスの情報で示されるソースバケット以後のＡＶストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、前記アドレスは決定される請求の範囲第２項記載の情報処理装置。

６． 前記第３のＡＶストリームが記録媒体上で所定の大きさ以上の連続領域に配置されるように、前記第３のＡＶストリームが生成される請求の範囲第２項記載の情報処理装置。

７． 第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成すると共に、前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り換わるタイミングにおける前記第１のＡＶストリームのソースバケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り換わるタイミングにおける前記第２のＡＶストリームのソースバケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含む情報処理方法。

８． 第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成すると共に、前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り換わるタイミングにおける前記第１のＡＶストリームのソースバケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り換わるタイミングにおける前記第２のＡＶストリームのソースバケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

9. 第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、前記第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースバケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースバケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップをコンピュータに実行させるプログラム。

10. 第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームを記録媒体から読み出す第1の読出手段と、

前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースバケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースバケットのアドレスの情報から構成されるアドレス情報を前記記録媒体から読み出す第2の読出手段と、

前記第2の読出手段により読み出された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出手段により読み出された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えて再生する再生手段とを含む情報処理装置。

11. 第1のAVストリーム、第2のAVストリーム、又は、第3のAVストリームの記録媒体からの読み出しを制御する第1の読出制御ステップと、

前記第3のAVストリームの関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースバケットのアドレスの情報と、前記第3のAVストリ

ームから前記第 2 の A V ストリームに再生が切り換わるタイミングにおける前記第 2 の A V ストリームのソースバケットのアドレスの情報から構成されるアドレス情報の前記記録媒体からの読み出しを制御する第 2 の読出制御ステップと、

前記第 2 の読出制御ステップの処理で読み出しが制御された前記第 3 の A V ストリームに関連する情報に基づいて、前記第 1 の読出制御ステップの処理で読み出しが制御された前記第 1 の A V ストリームから前記第 3 の A V ストリームへ再生を切り換え、前記第 3 の A V ストリームから前記第 2 の A V ストリームへ再生を切り換えて再生する再生ステップとを含む情報処理方法。

12. 第 1 の A V ストリーム、第 2 の A V ストリーム、又は、第 3 の A V ストリームの記録媒体からの読み出しを制御する第 1 の読出制御ステップと、

前記第 3 の A V ストリームに関連する情報として、前記第 1 の A V ストリームから前記第 3 の A V ストリームに再生が切り換わるタイミングにおける前記第 1 の A V ストリームのソースバケットのアドレスの情報と、前記第 3 の A V ストリームから前記第 2 の A V ストリームに再生が切り換わるタイミングにおける前記第 2 の A V ストリームのソースバケットのアドレスの情報から構成されるアドレス情報の前記記録媒体からの読み出しを制御する第 2 の読出制御ステップと、

前記第 2 の読出制御ステップの処理で読み出しが制御された前記第 3 の A V ストリームに関連する情報に基づいて、前記第 1 の読出制御ステップの処理で読み出しが制御された前記第 1 の A V ストリームから前記第 3 の A V ストリームへ再生を切り換え、前記第 3 の A V ストリームから前記第 2 の A V ストリームへ再生を切り換えて再生する再生ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

13. 第 1 の A V ストリーム、第 2 の A V ストリーム、又は、第 3 の A V ストリームの記録媒体からの読み出しを制御する第 1 の読出制御ステップと、

前記第 3 の A V ストリームに関連する情報として、前記第 1 の A V ストリームから前記第 3 の A V ストリームに再生が切り換わるタイミングにおける前記第 1 の A V ストリームのソースバケットのアドレスの情報と、前記第 3 の A V ストリームから前記第 2 の A V ストリームに再生が切り換わるタイミングにおける前記第 2 の A V ストリームのソースバケットのアドレスの情報から構成されるアドレ

ス情報の前記記録媒体からの読み出しを制御する第2の読出制御ステップと、

前記第2の読出制御ステップの処理で読み出しが制御された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出制御ステップの処理で読み出しが制御された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えて再生する再生ステップとをコンピュータに実行させるプログラム。

14. 第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、前記第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームと、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースバケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースバケットのアドレスの情報から構成されるアドレス情報が記録されている記録媒体。

補正書の請求の範囲

[2001年8月2日(02.08.01)国際事務局受理：出願当初の請求の範囲
10, 11, 12 及び 13 は補正された；他の請求の範囲は変更なし。(3頁)]

9. 第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、前記第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームを生成すると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップをコンピュータに実行させるプログラム。

10. (補正後) 第1のAVストリーム、第2のAVストリーム、及び当該第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームが記録されると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録された記録媒体を再生する再生手段と、

前記第2の読出手段により読み出された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出手段により読み出された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えるよう前記再生手段を制御する制御手段とを含む情報処理装置。

11. (補正後) 第1のAVストリーム、第2のAVストリーム、及び当該第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切

り換えられるとき再生される第3のAVストリームが記録されると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録された記録媒体を再生する再生ステップと、

前記第2の読出制御ステップの処理で読み出しが制御された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出制御ステップの処理で読み出しが制御された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えて再生するよう再生を制御する制御ステップとを含む情報処理方法。

12. (補正後) 第1のAVストリーム、第2のAVストリーム、及び当該第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームが記録されると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録された記録媒体を再生する再生ステップと、

前記第2の読出制御ステップの処理で読み出しが制御された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出制御ステップの処理で読み出しが制御された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えて再生するよう再生を制御する制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

13. (補正後) 第1のAVストリーム、第2のAVストリーム、及び当該第

1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームが記録されると共に、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録された記録媒体を再生する再生ステップと、

前記第2の読出制御ステップの処理で読み出しが制御された前記第3のAVストリームに関連する情報に基づいて、前記第1の読出制御ステップの処理で読み出しが制御された前記第1のAVストリームから前記第3のAVストリームへ再生を切り換え、前記第3のAVストリームから前記第2のAVストリームへ再生を切り換えて再生するよう再生を制御する制御ステップとをコンピュータに実行させるプログラム。

14. 第1のAVストリームから第2のAVストリームへ連続的に再生されるように指示された場合、前記第1のAVストリームの所定の部分と前記第2のAVストリームの所定の部分から構成され、前記第1のAVストリームから前記第2のAVストリームに再生が切り換えられるとき再生される第3のAVストリームと、前記第3のAVストリームに関連する情報として、前記第1のAVストリームから前記第3のAVストリームに再生が切り換わるタイミングにおける前記第1のAVストリームのソースパケットのアドレスの情報と、前記第3のAVストリームから前記第2のAVストリームに再生が切り換わるタイミングにおける前記第2のAVストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録されている記録媒体。

1/100

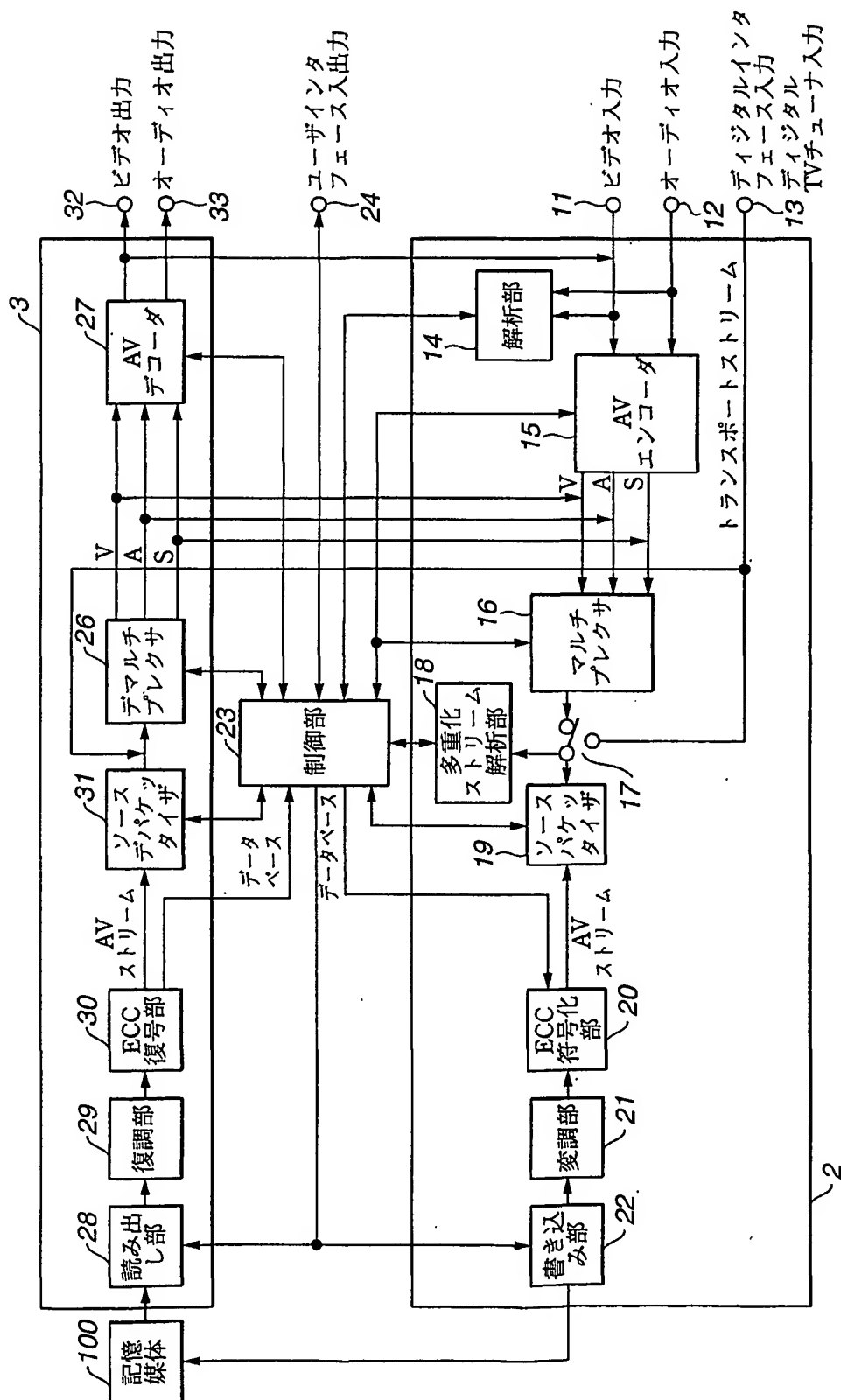
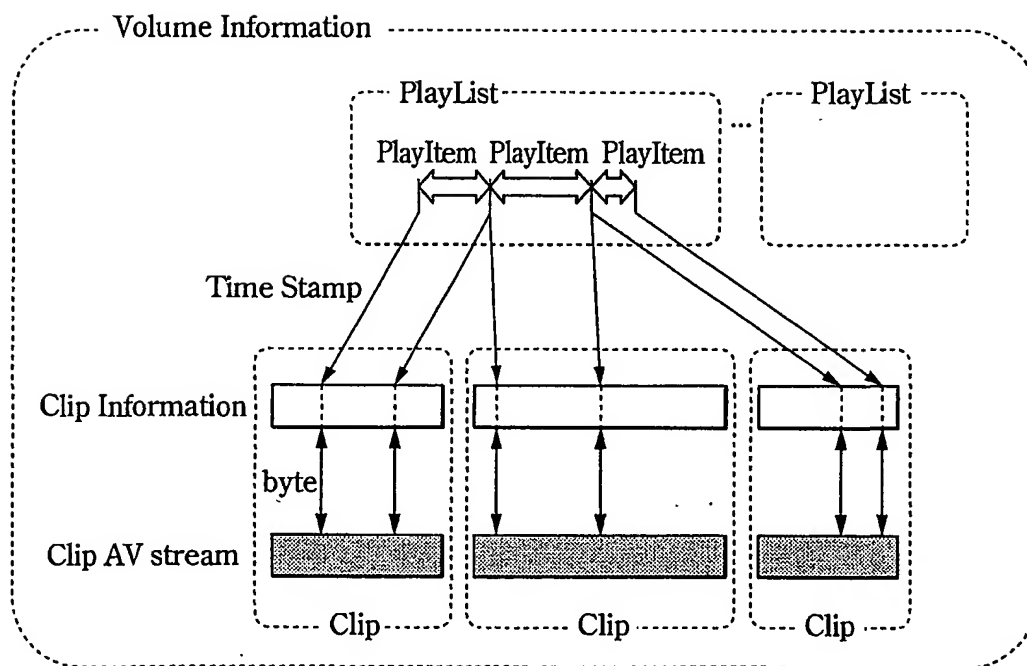


FIG.1

2/100

**FIG.2**

3/100

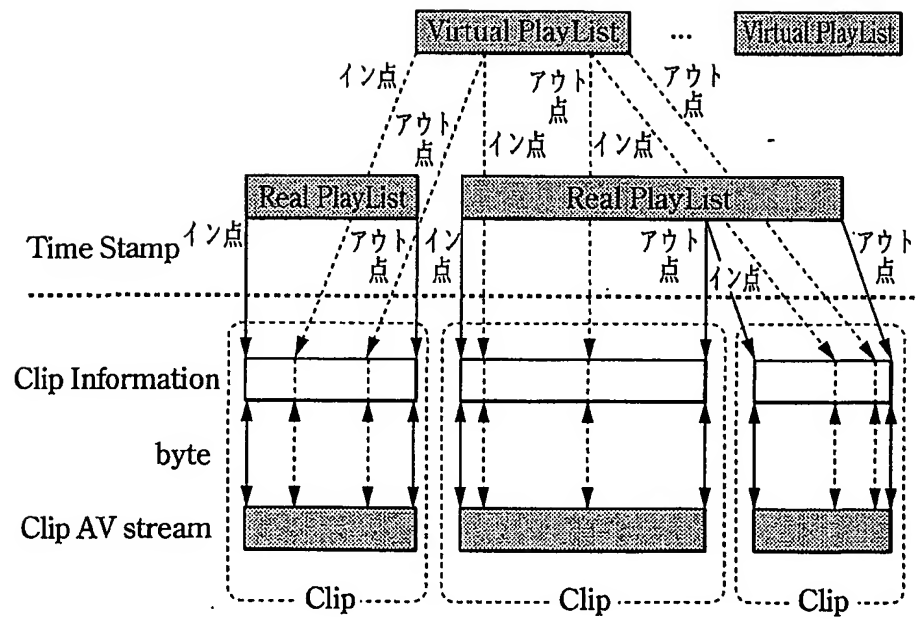


FIG.3

4/100

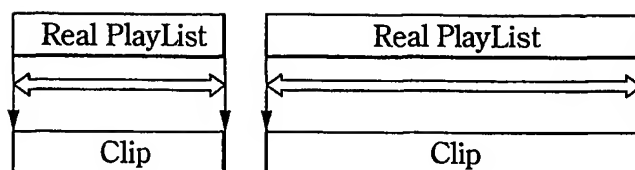


FIG.4A

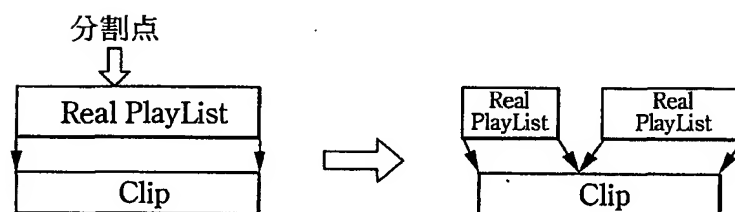


FIG.4B

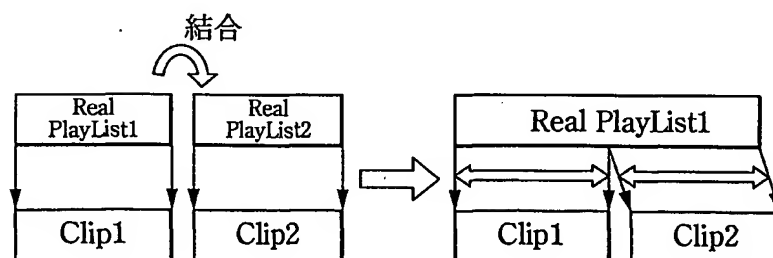


FIG.4C

5/100

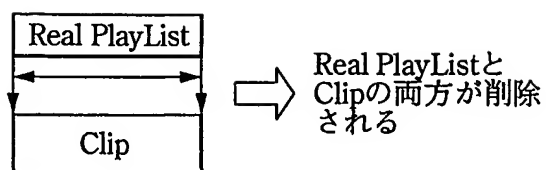


FIG.5A

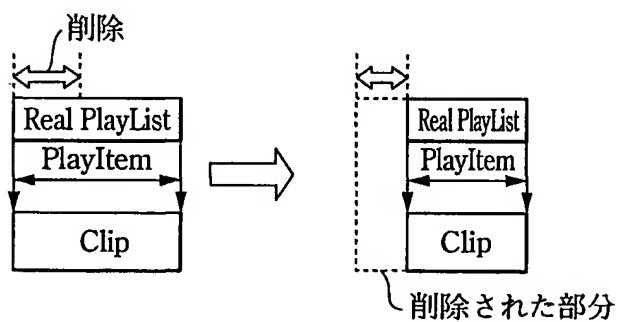


FIG.5B

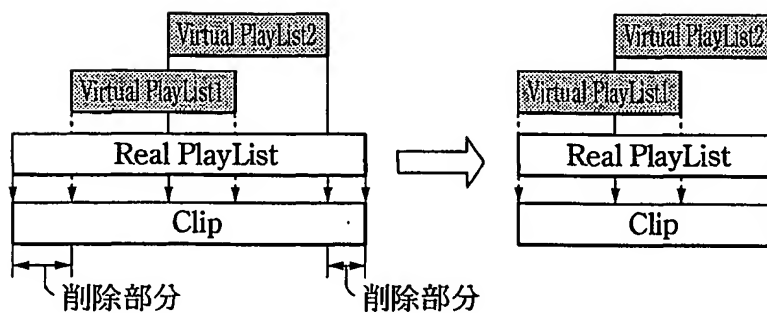
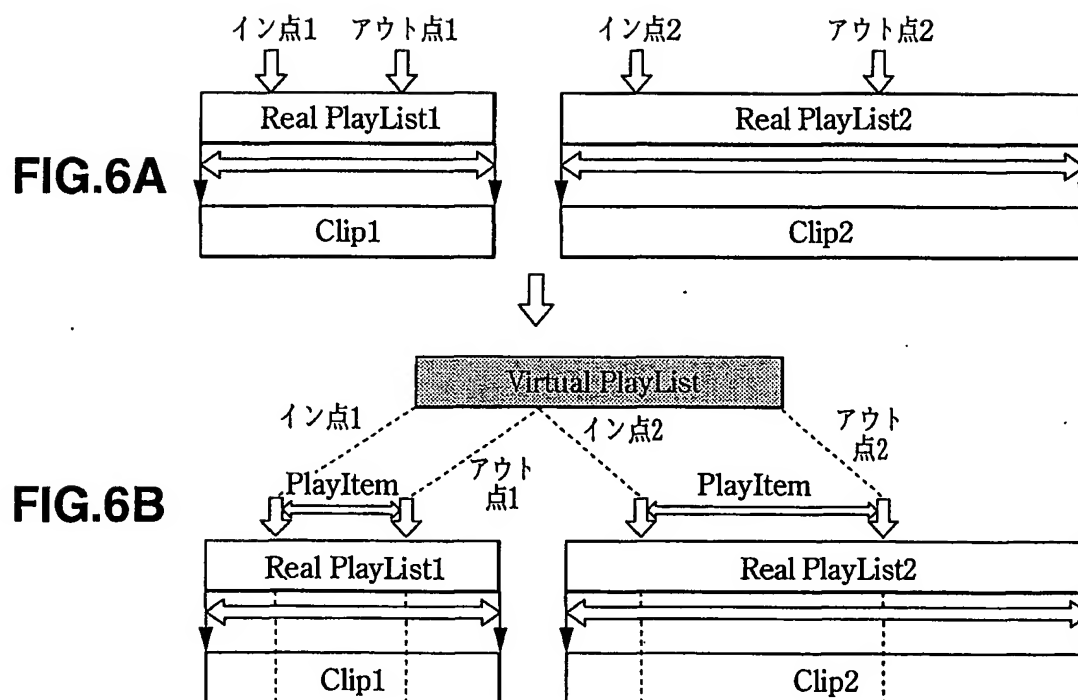


FIG.5C

6/100



7/100

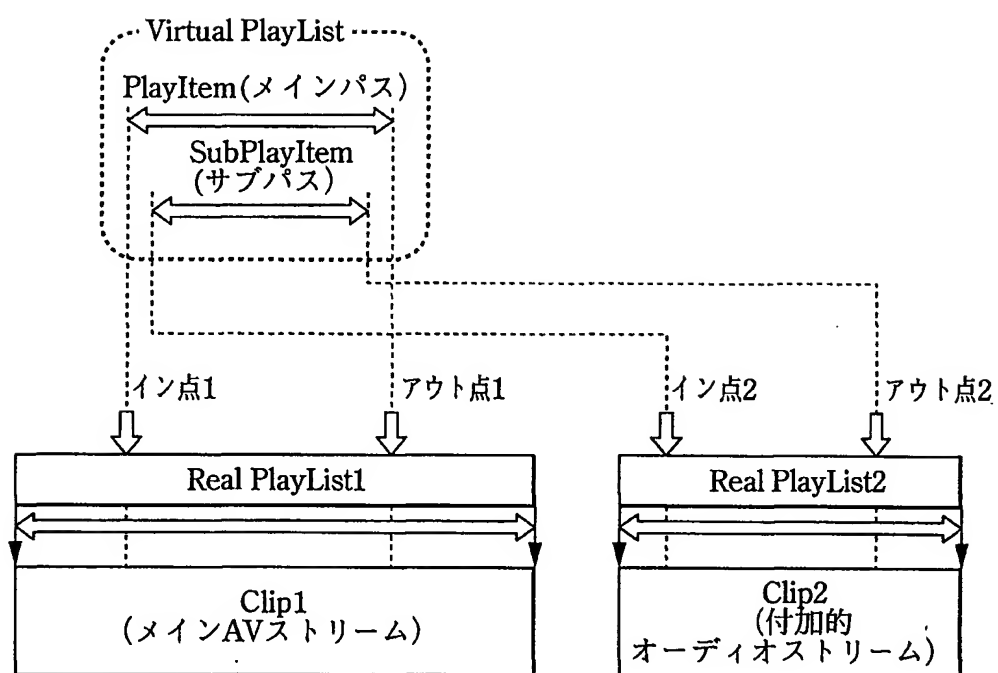


FIG.7

8/100

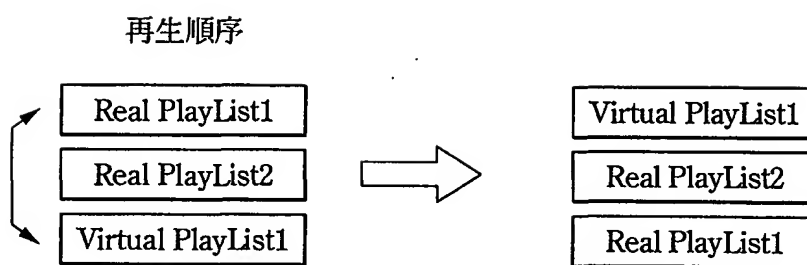


FIG.8

9/100

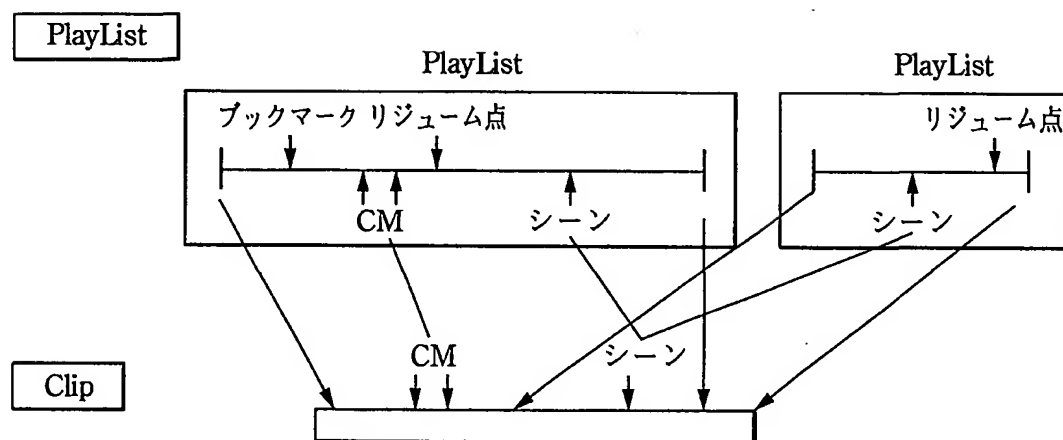


FIG.9

10/100

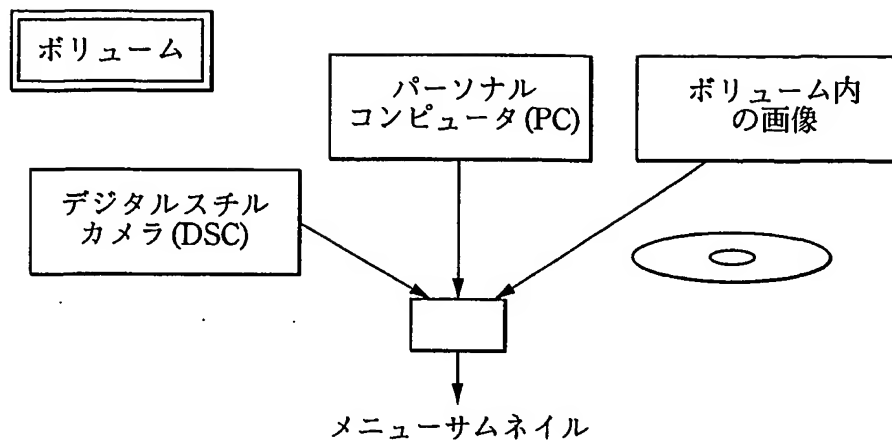


FIG.10

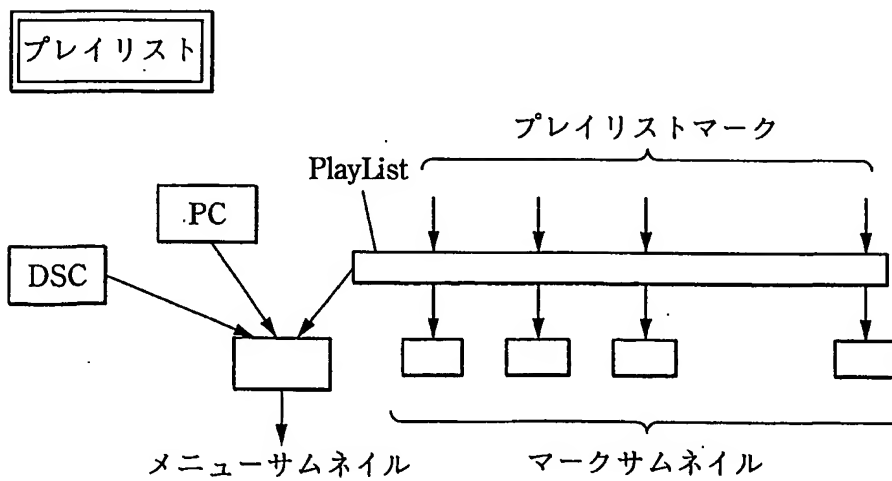


FIG.11

11/100

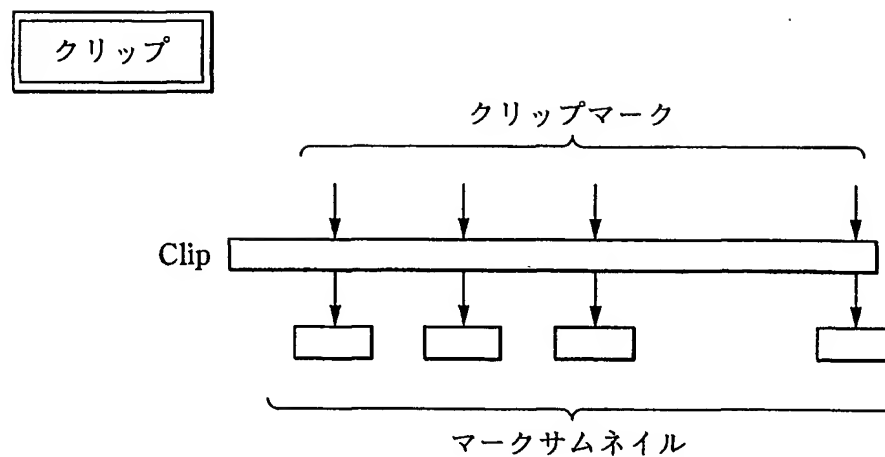


FIG.12

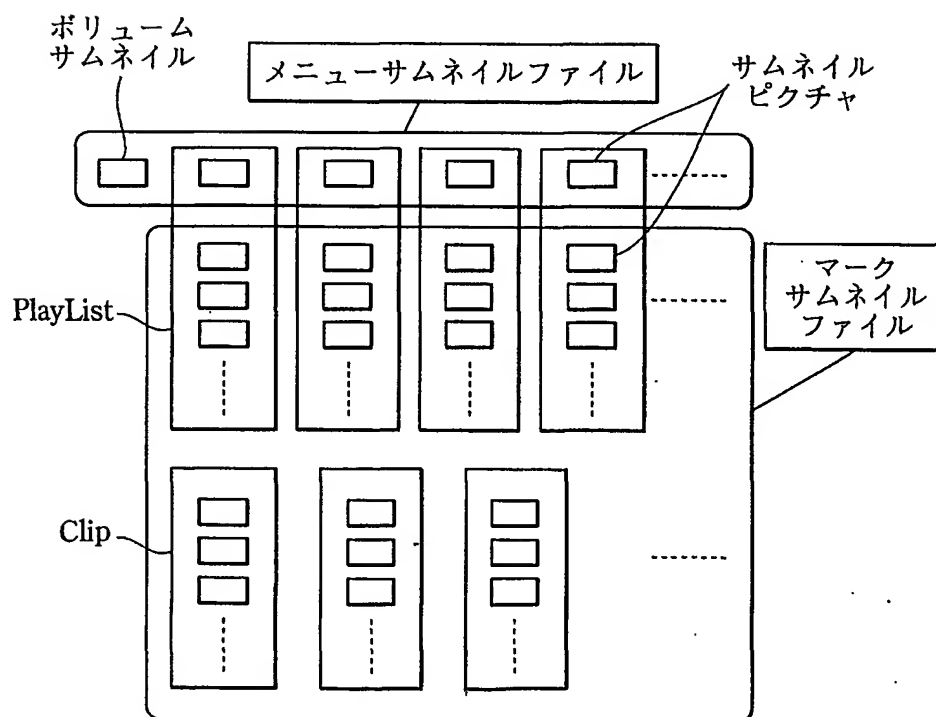


FIG.13

12/100

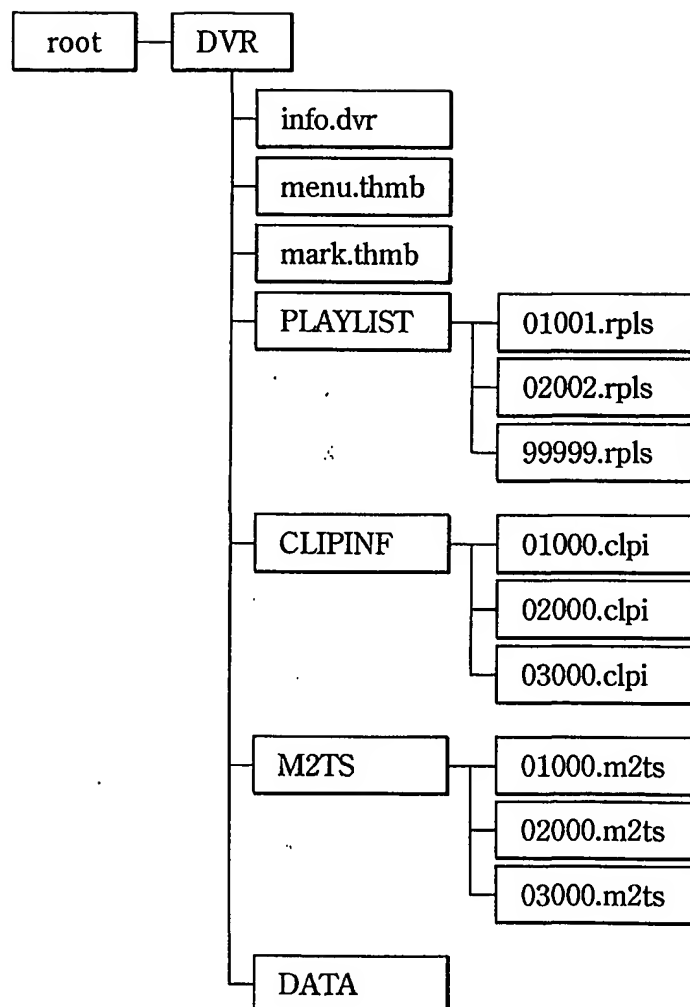


FIG.14

13/100

シンタクス	バイト数	略号
info.dvr {		
TableOfPlayLists_Start_address	32	uimbsf
MakersPrivateData_Start_address	32	uimbsf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.15

14/100

シンタクス	バイト数	略号
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

シンタクス	バイト数	略号
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

16/100

シンタクス	バイト数	略号
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.18

17/100

値	キャラクタ文字符号化
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

FIG.19

18/100

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

FIG.20

19/100

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

FIG.21

20/100

シンタクス	バイト数	略号
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_ size*1024*8	
}		
}		
}		

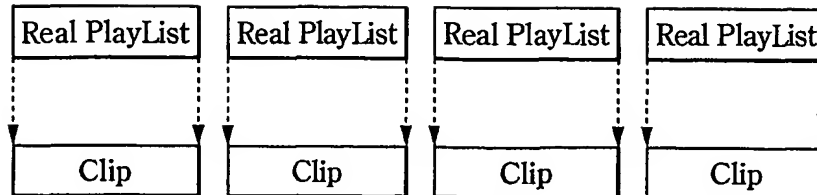
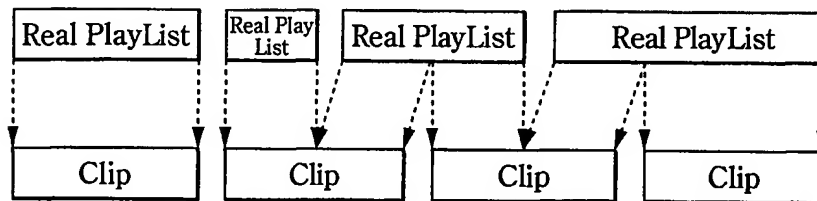
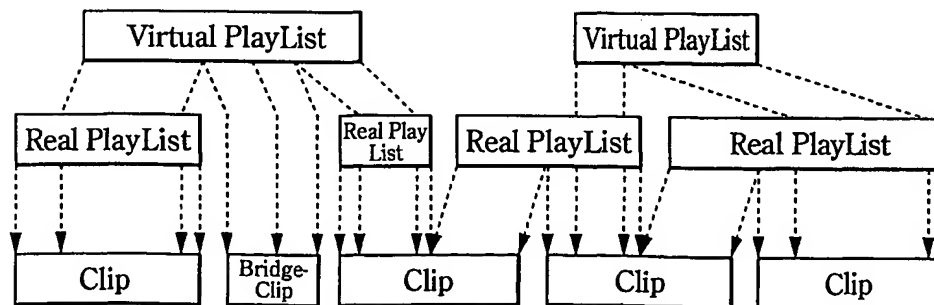
FIG.22

21/100

シンタクス	バイト数	略号
xxxxxx.rpls / yyyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

22/100

**FIG.24A****FIG.24B****FIG.24C**

23/100

シンタクス	バイト数	略号
PlayList(){		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

FIG.25

24/100

PlayList_type	意味
0	AV記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

FIG.26

25/100

シンタクス	バイト数	略号
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.27

26/100

write_protect_flag	意味
0b	そのPlayListを自由に消去しても良い。
1b	write_protect_flagを除いてそのPlayListの内容は、消去および変更されるべきではない。

FIG.28A

is_played_flag	意味
0b	そのPlayListは、記録されてから一度も再生されたことがない。
1b	PlayListは、記録されてから一度は再生された。

FIG.28B

archive	意味
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

FIG.28C

27/100

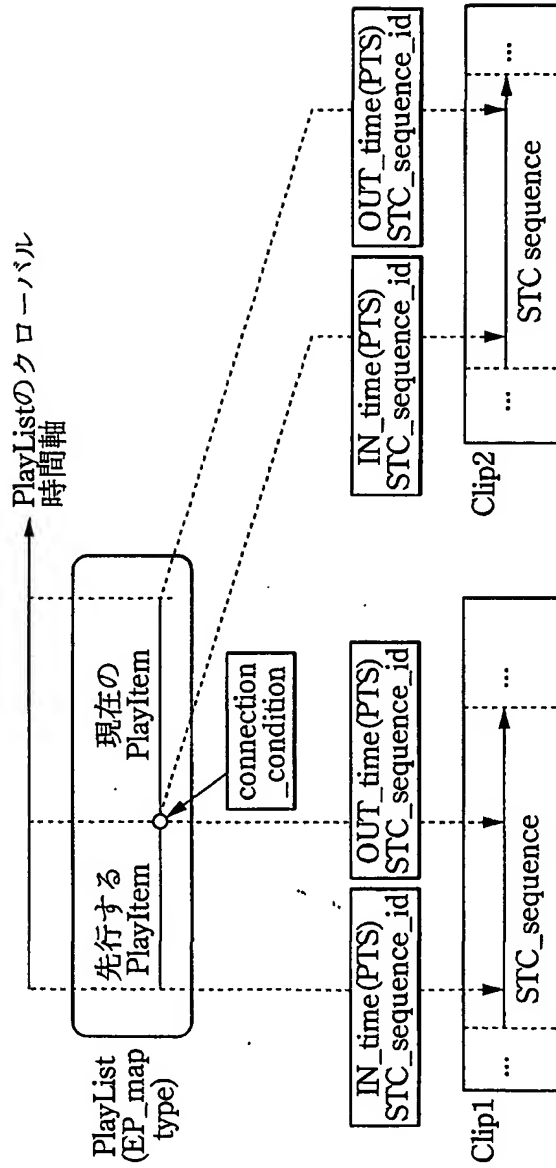


FIG.29

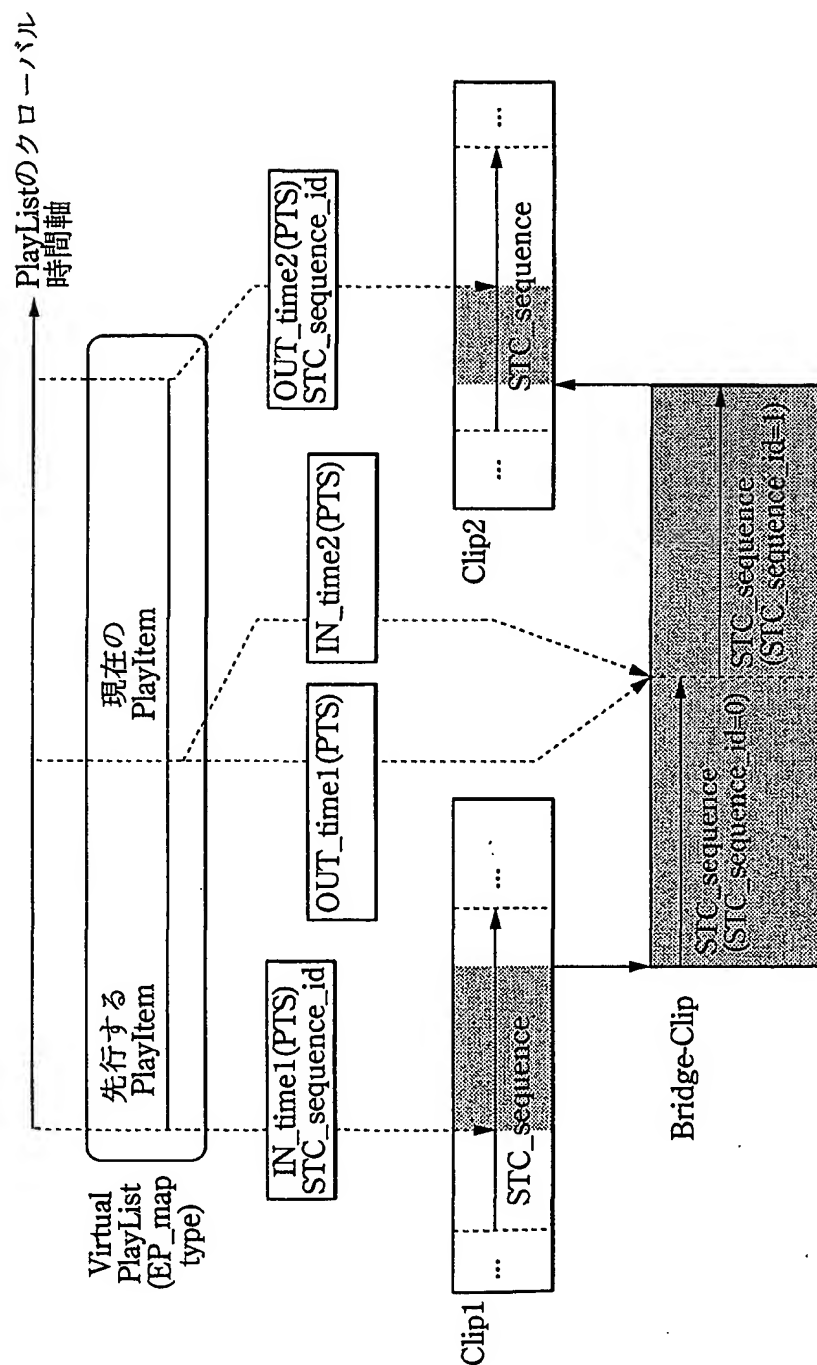


FIG.30

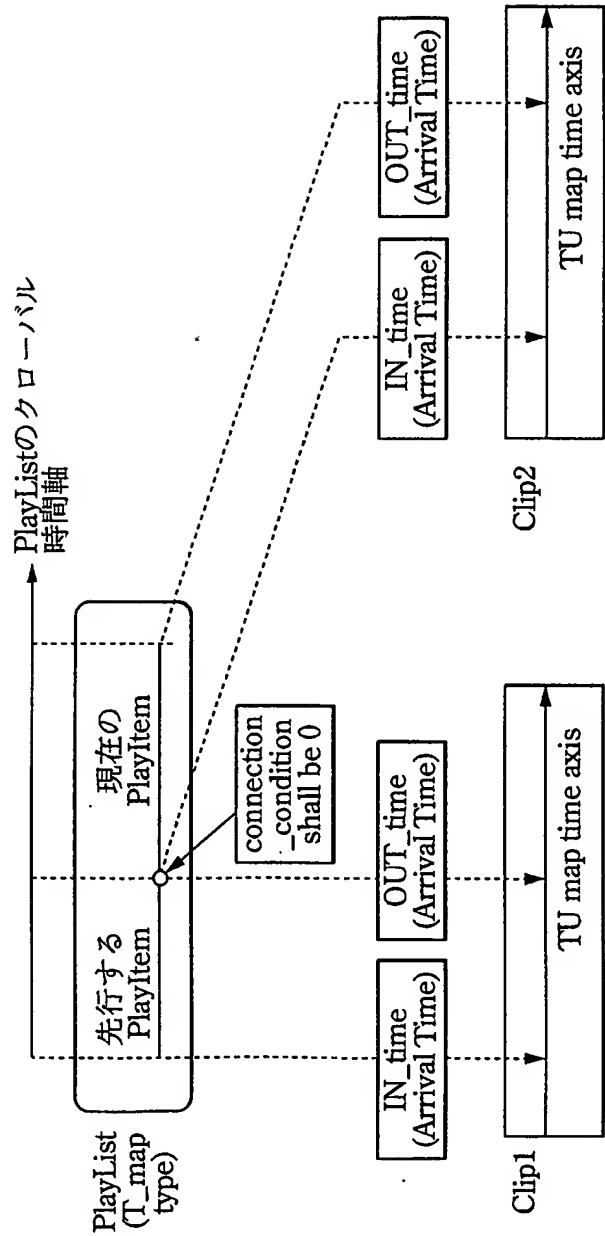


FIG.31

シンタクス	バイト数	略号
PlayItem(){		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (connection_condition=='10'){		
BridgeSequenceInfo()		
}		
}		
}		

FIG.32

31/100

CPI_type in the PlayList()	IN_timeのセマンティクス
EP_map type	IN_timeは、PlayItemの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	<p>IN_timeは、TU_map_time_axis上の時刻でなければならない。 かつ、IN_timeは、time_unitの精度に丸めて表さなければならない。IN_timeは、次に示す等式により計算される。</p> $IN_time = TU_start_time \% 2^{32}$

FIG.33

32/100

CPI_type in the PlayList()	OUT_timeのセマンティクス
EP_map type	<p>OUT_timeは、次に示す等式により計算される Presentation_end_TSの値の上位32ビットを示さなければならない。</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>ここで、</p> <p>PTS_outは、PlayItemの中で最後のプレゼンテーションユニットに対応する33ビット長のPTSである。</p> <p>AU_durationは、最後のプレゼンテーションユニットの90kHz単位の表示期間である。</p>
TU_map type	<p>OUT_timeは、TU_map_time_axis上の時刻でなければならない。かつ、OUT_timeは、time_unitの精度に丸めて表さなければならない。OUT_timeは、次に示す等式により計算される。</p> $\text{OUT_time} = \text{TU_start_time} \% 2^{32}$

FIG.34

33/100

connection _condition	意味
00	<ul style="list-style-type: none">・ 先行するPlayItemと現在のPlayItemの接続は、シームレス再生の保証がなされていない。・ PlayListのCPI_typeがTU_map typeである場合、connection_conditionは、この値をセットされねばならない。
01	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ 先行するPlayItemと現在のPlayItemは、システムタイムベース（STCベース）の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ この状態は、Virtual PlayListに対してだけ許される。・ 先行するPlayItemと現在のPlayItemとの接続は、シームレス再生の保証がなされている。・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用して接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。
11	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ 先行するPlayItemと現在のPlayItemは、シームレス再生の保証がなされている。・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用しないで接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。

FIG.35

34/100

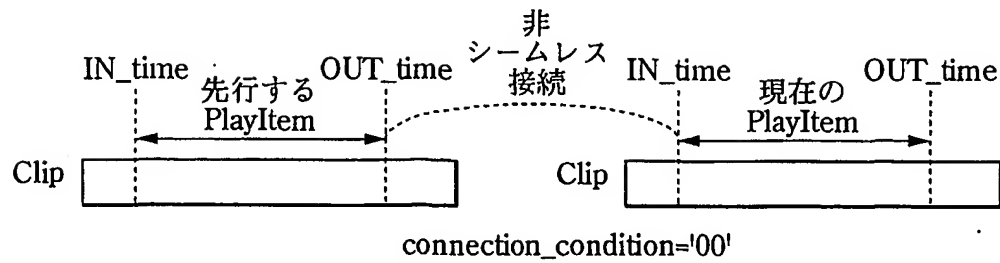


FIG.36A

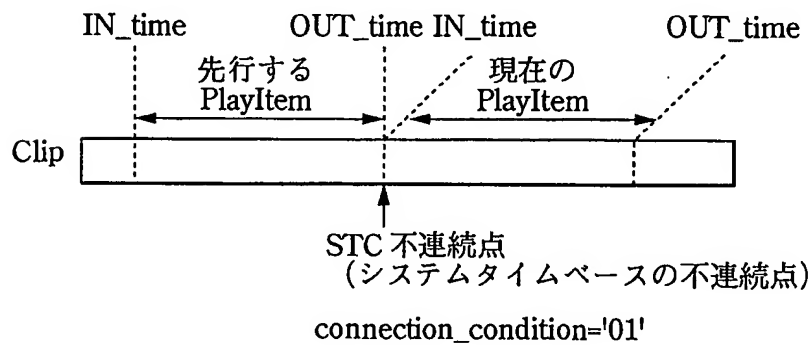


FIG.36B

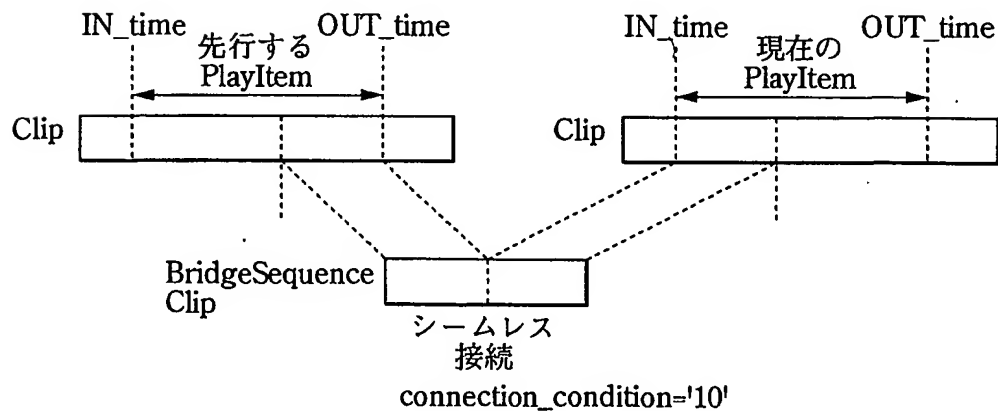


FIG.36C

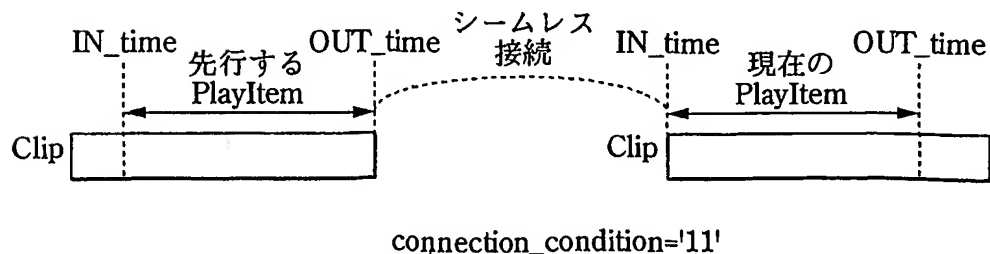


FIG.36D

35/100

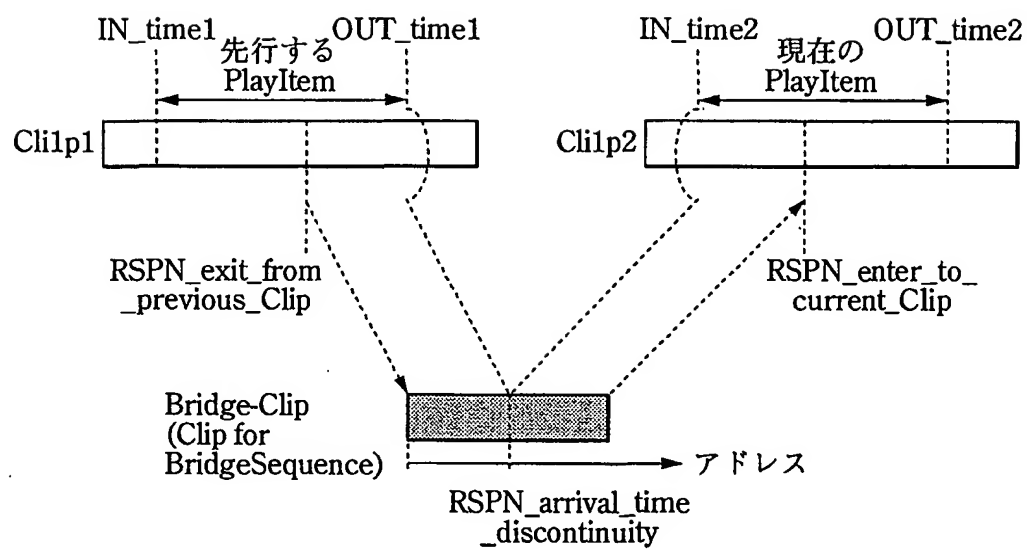


FIG.37

36/100

シンタクス	バイト数	略号
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

37/100

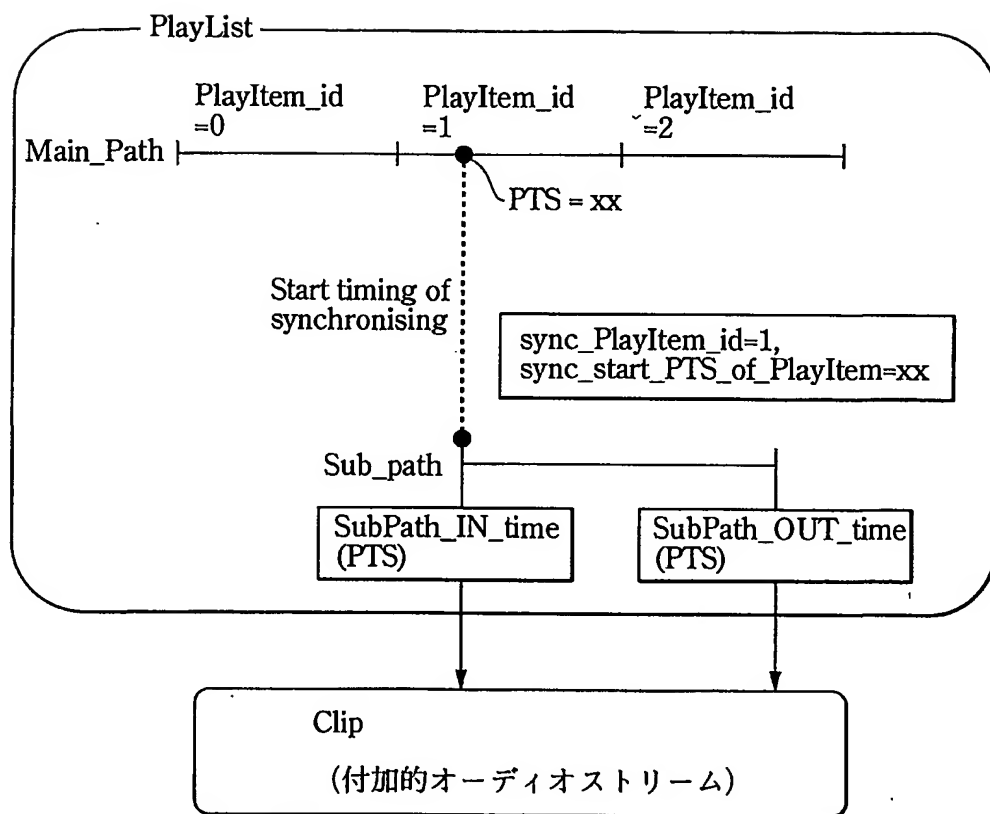


FIG.39

38/100

シンタクス	バイト数	略号
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

FIG.40

39/100

SubPath_type	意味
0x00	Auxiliary audio steam path
0x01-0xff	reserved

FIG.41

40/100

シンタクス	バイト数	略号
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.42

41/100

Mark_type	意味	コメント
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

42/100

CPI_type in the PlayList()	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	<p>mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次 に示す等式により計算される。</p> $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

FIG.44

43/100

シンタクス	バイト数	略号
zzzzz.clpi {		
STC_Info_Start_address	32	uimbsf
ProgramInfo_Start_address	32	uimbsf
CPI_Start_address	32	uimbsf
ClipMark_Start_address	32	uimbsf
MakersPrivateData_Start_address	32	uimbsf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.45

44/100

シンタクス	バイト数	略号
ClipInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

FIG.46

45/100

Clip_stream_type	意味
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2-255	Reserved

FIG.47

46/100

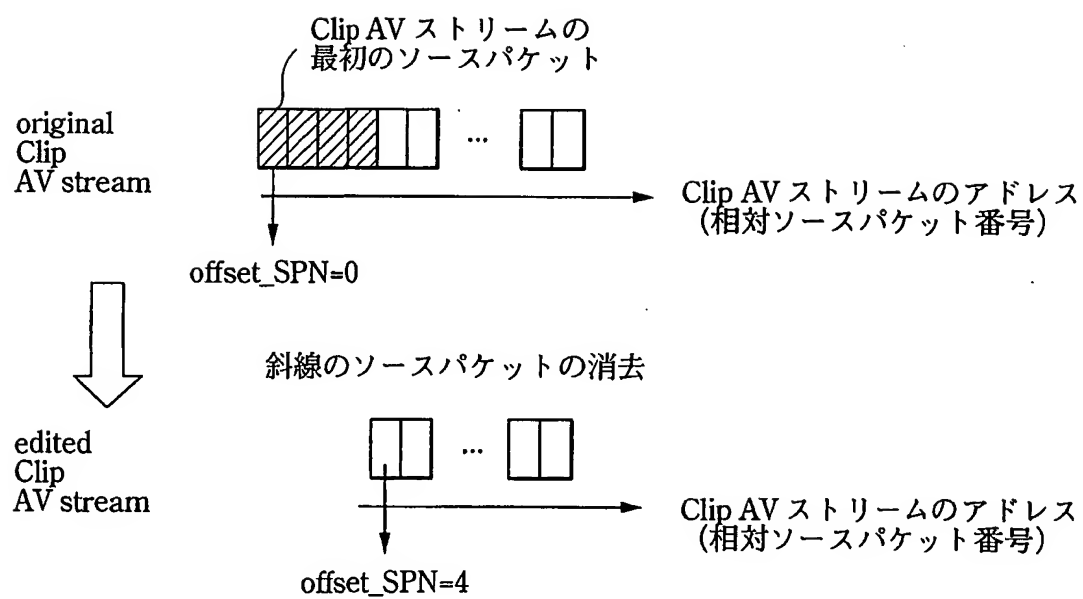


FIG.48

47/100

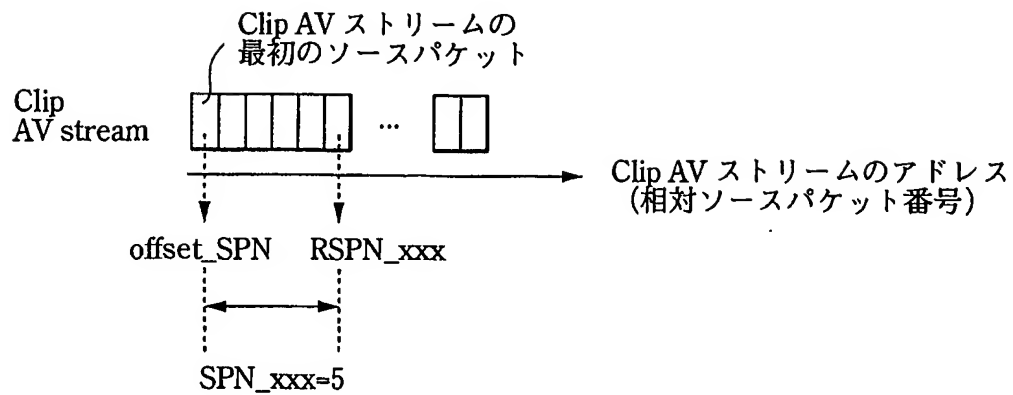


FIG.49

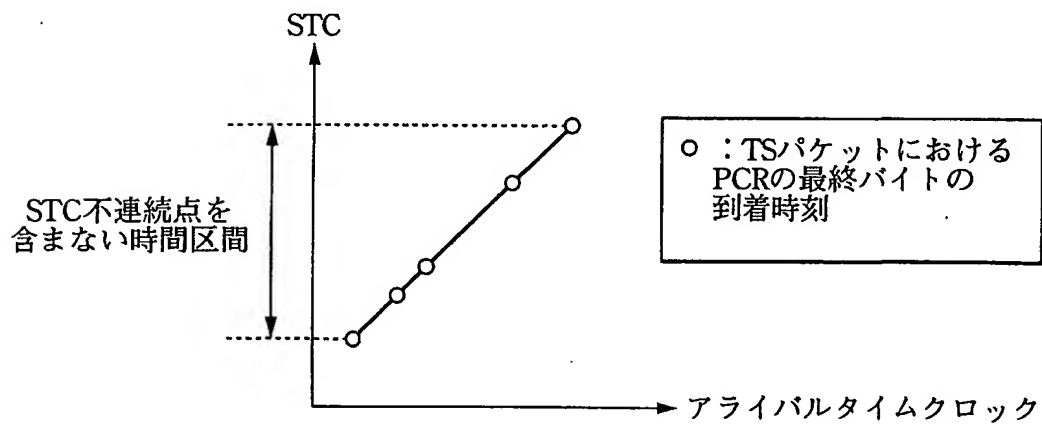


FIG.50A

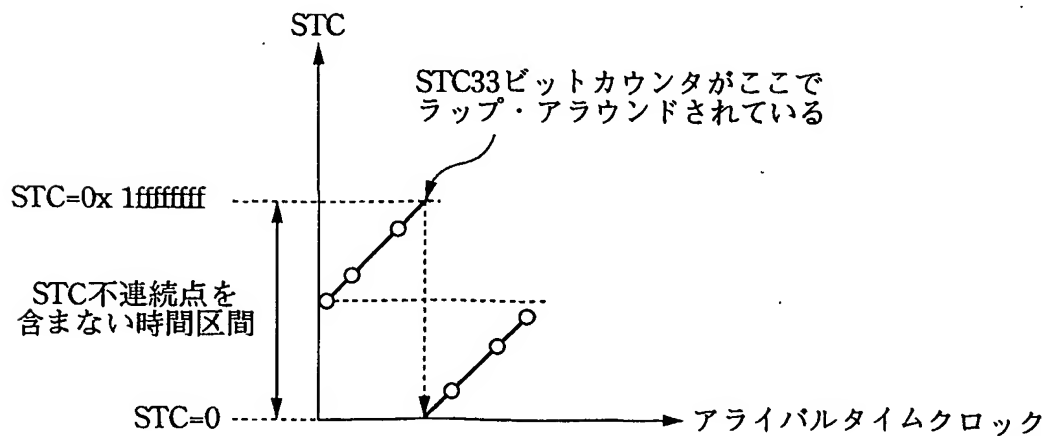


FIG.50B

48/100

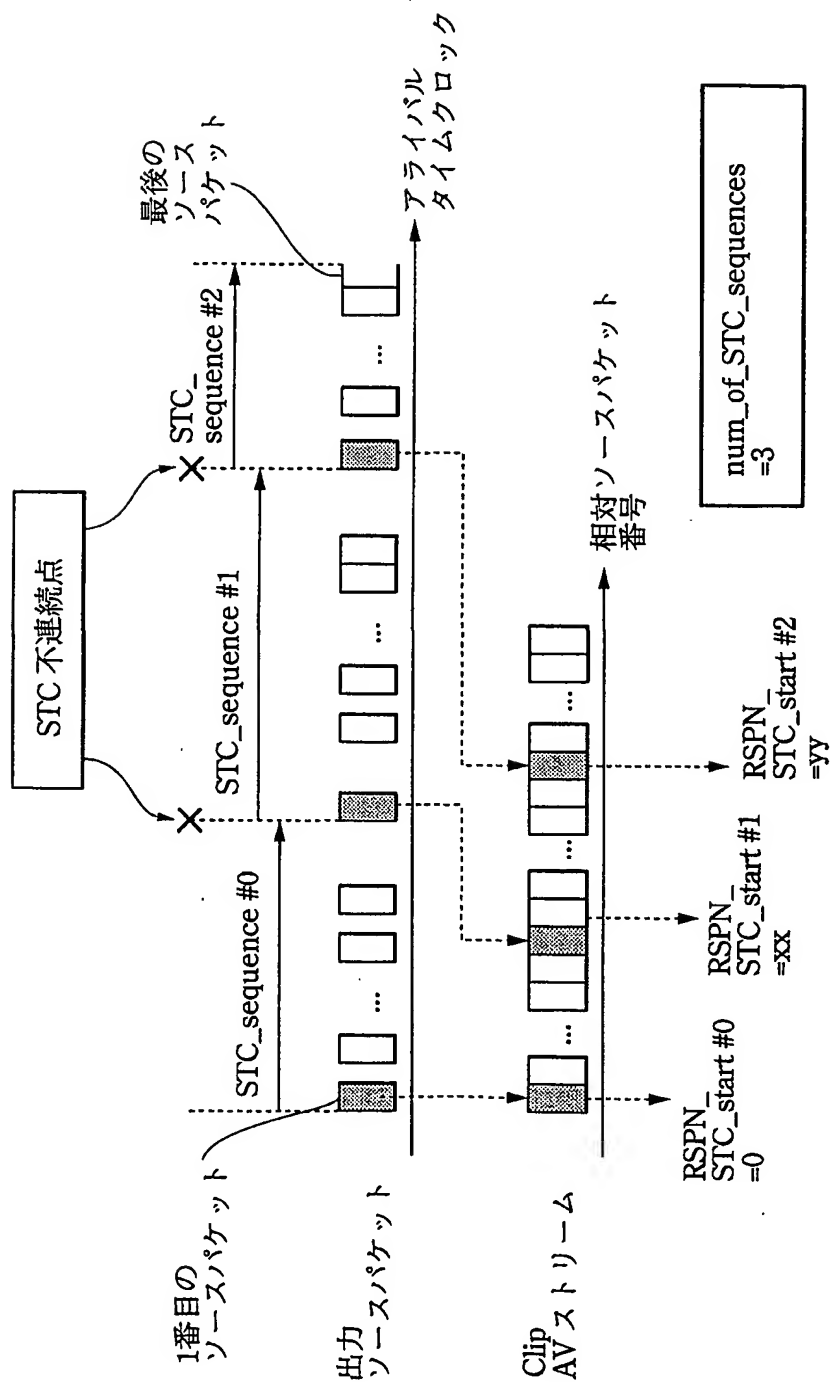
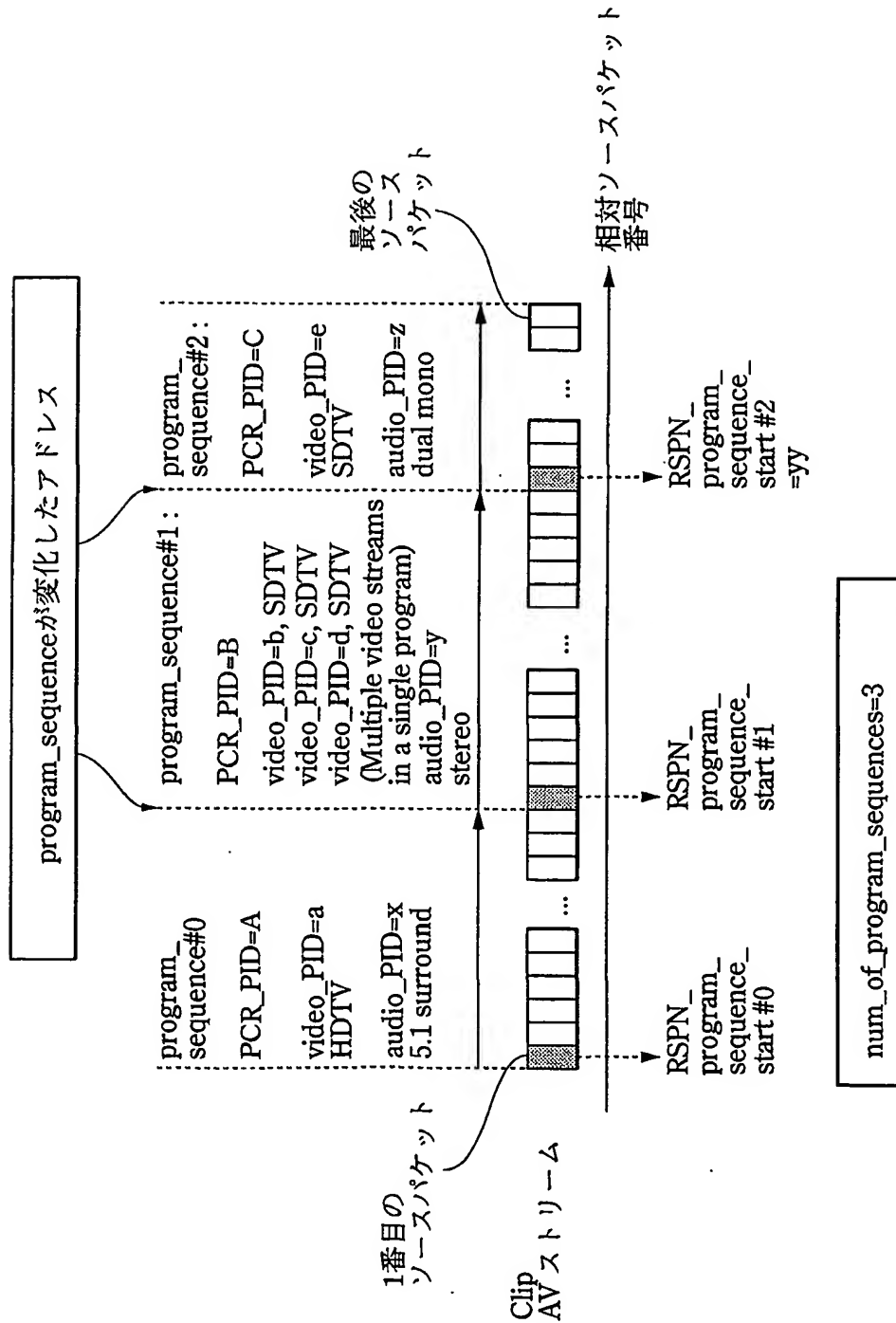


FIG.51

49/100

シンタクス	バイト数	略号
STC_Info(){		
version_number	8*4	bslbf
length	32	uimbsf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimbsf
for (STC_sequence_id=0; STC_sequence_id<num_of_STC_sequences; STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimbsf
}		
}		
}		

FIG.52



51/100

シンタクス	バイト数	略号
ProgramInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

FIG.54

シンタクス	バイト数	略号
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

FIG.55

53/100

video_format	意味
0	480i
1	576i
2	480p (including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

FIG.56

frame_rate	意味
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

FIG.57

54/100

display_aspect_ratio	意味
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

FIG.58

55/100

シンタクス	バイト数	略号
AudioCodingInfo() {		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

FIG.59

56/100

audio_coding	意味
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

FIG.60

57/100

audio_component_type	意味
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

FIG.61

sampling_frequency	意味
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

FIG.62

58/100

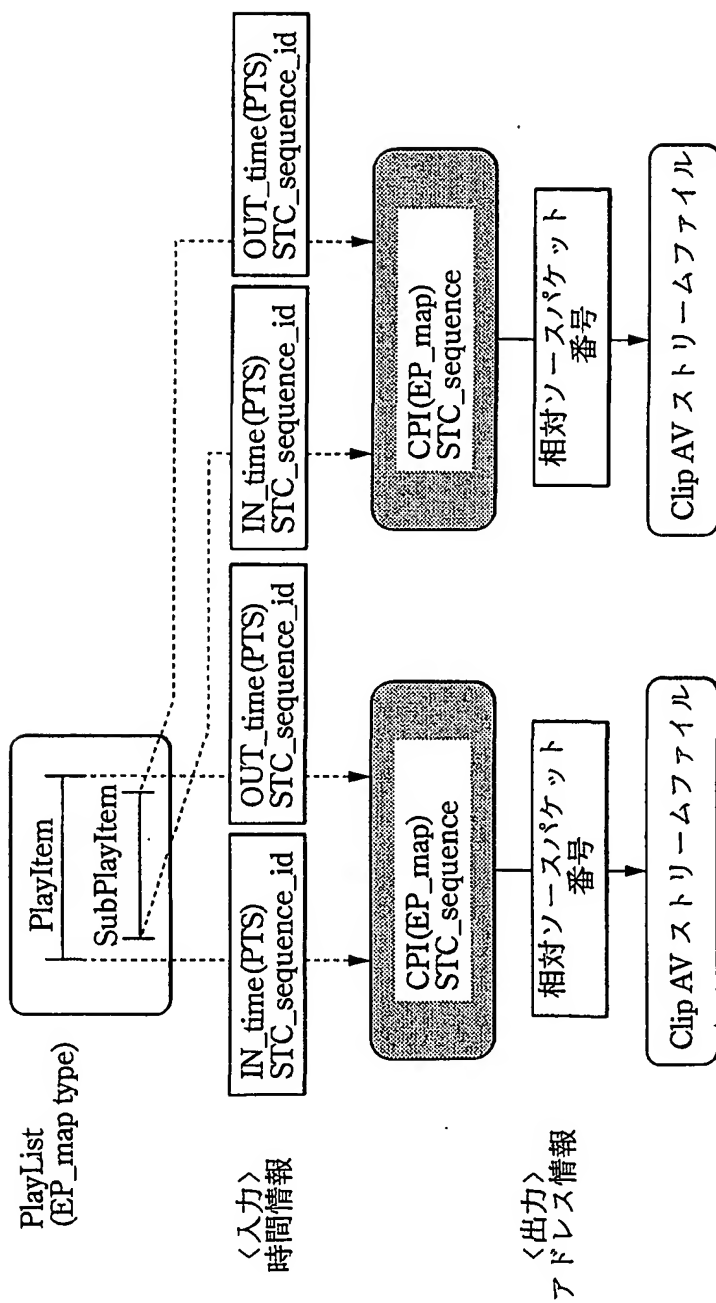


FIG.63

59/100

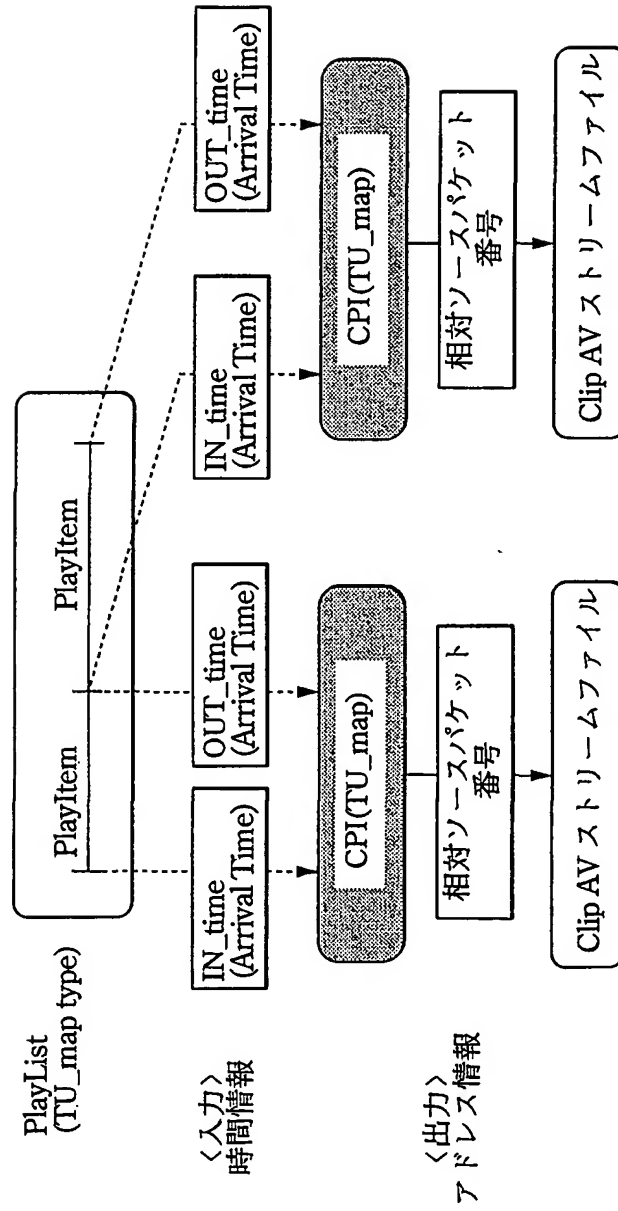


FIG.64

60/100

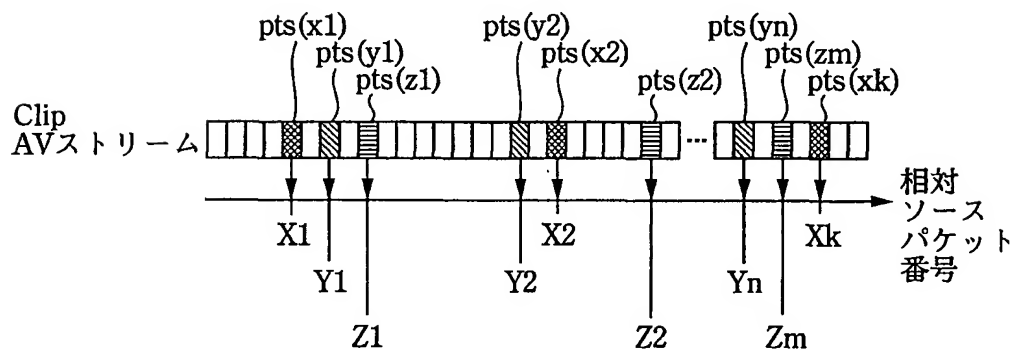
シンタクス	バイト数	略号
CPI(){		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

FIG.65

61/100

CPI_type	意味
0	EP map type
1	TU map type

FIG.66



- : シーケンスヘッダvideo_PID=xの第1バイト目を含むソースパケット
- ▨ : シーケンスヘッダvideo_PID=yの第1バイト目を含むソースパケット
- ▤ : シーケンスヘッダvideo_PID=zの第1バイト目を含むソースパケット

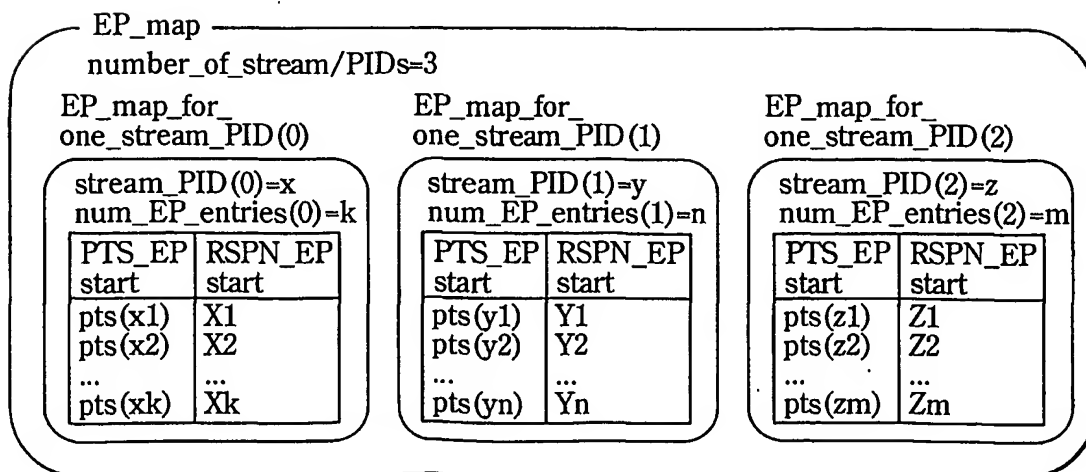
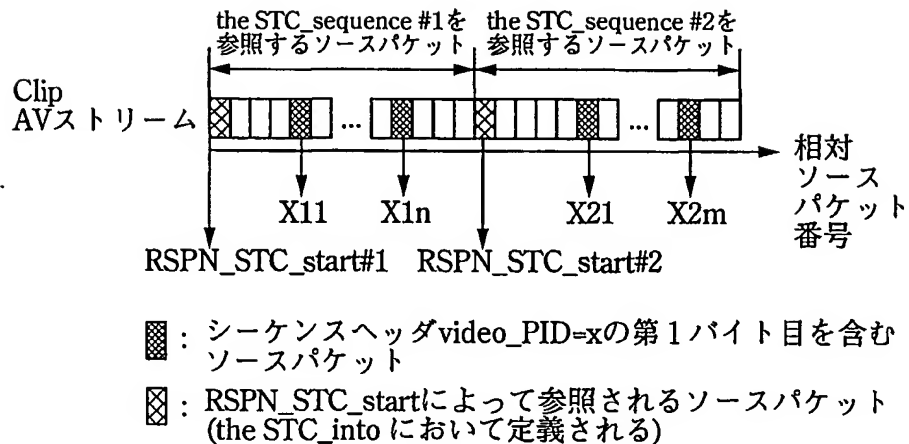


FIG.67

62/100



EP_map_for_one_stream_PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11) STC_sequence #1に属するデータ
...	...	
pts(x1n)	X1n) STC_sequence #2に属するデータ
...	...	
pts(x21)	X21) STC_sequence #2に属するデータ
...	...	
pts(x2m)	X2m	

boundary

RSPN_STC_start #2 < X21

FIG.68

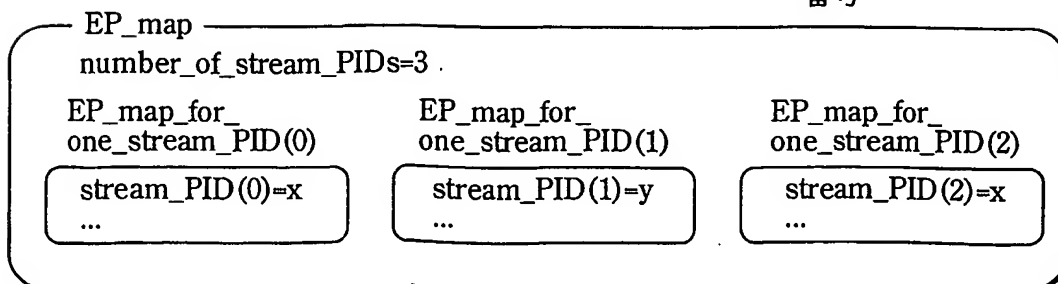
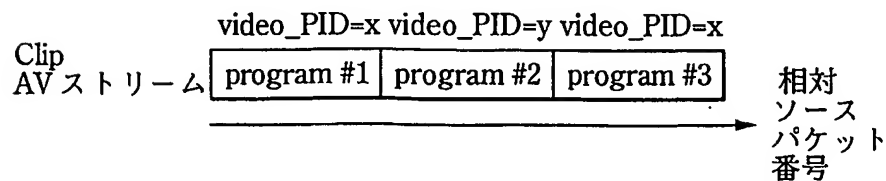


FIG.69

63/100

シンタクス	バイト数	略号
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

64/100

EP_type	意味
0	video
1	audio
2-15	reserved

FIG.71

65/100

シンタクス	バイト数	略号
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

FIG.72

66/100

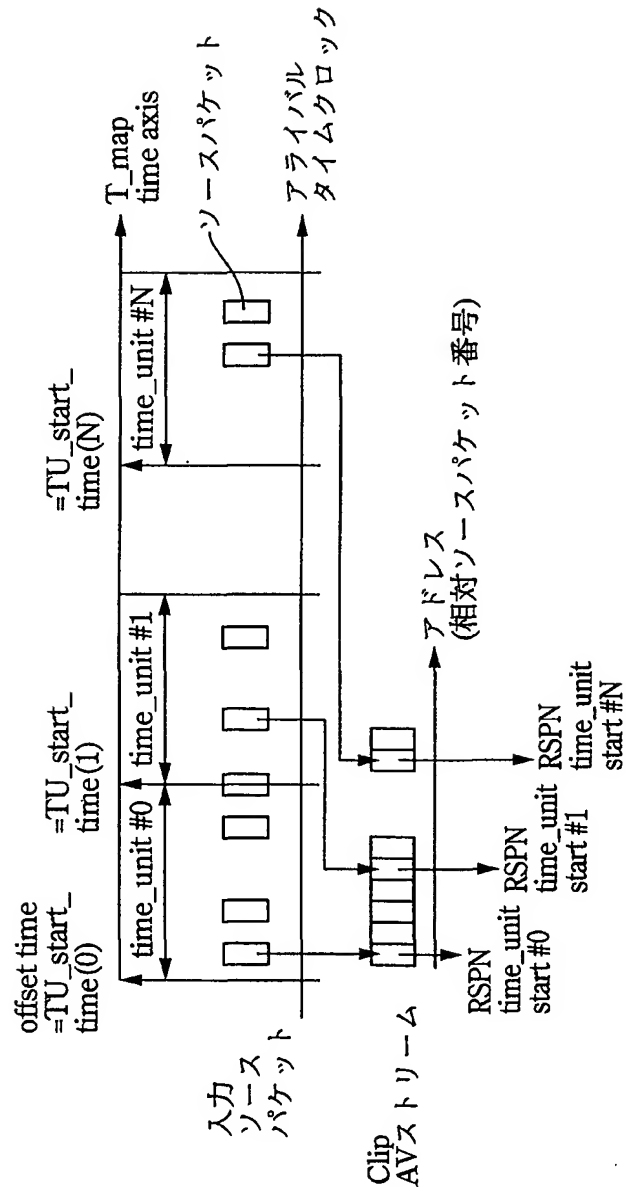


FIG.73

67/100

シンタクス	バイト数	略号
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

68/100

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.75

69/100

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーンチェンジポイントを示すマーク点
0x93-0xFF	reserved	

FIG.76

70/100

CPI_type in the CPI()	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

FIG.77

71/100

シンタクス	バイト数	略号
menu.thmb/mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for (i=0; i<N1; i++)		
padding_word	16	bslbf
}		

FIG.78

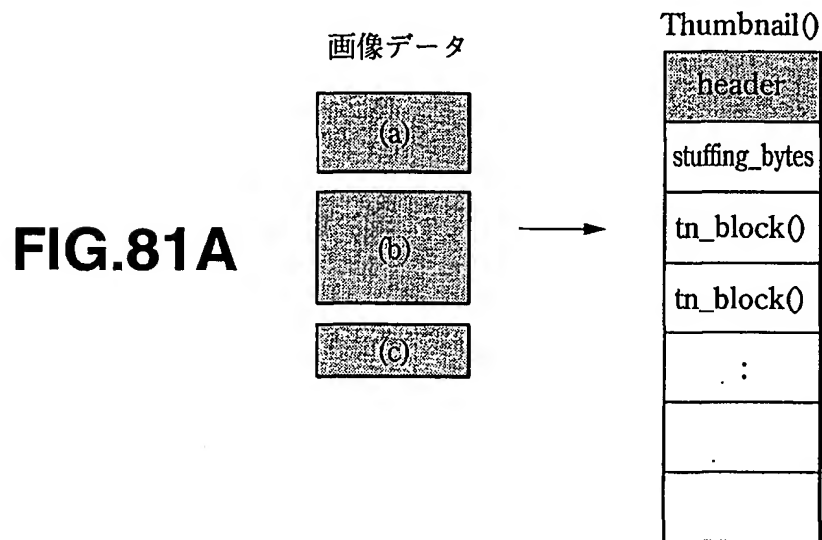
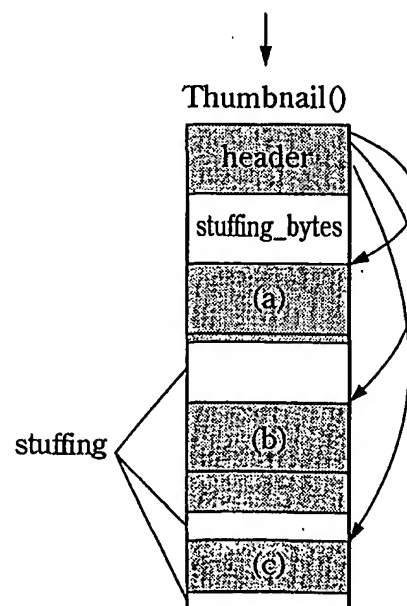
72/100

シンタクス	バイト数	略号
Thumbnail(){		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_ size*1024*8	
}		
}		
}		

FIG.79

73/100

Thumbnail_picture_format	意味
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

FIG.80**FIG.81 B**

74/100

DVR MPEG-2 トランスポートストリーム

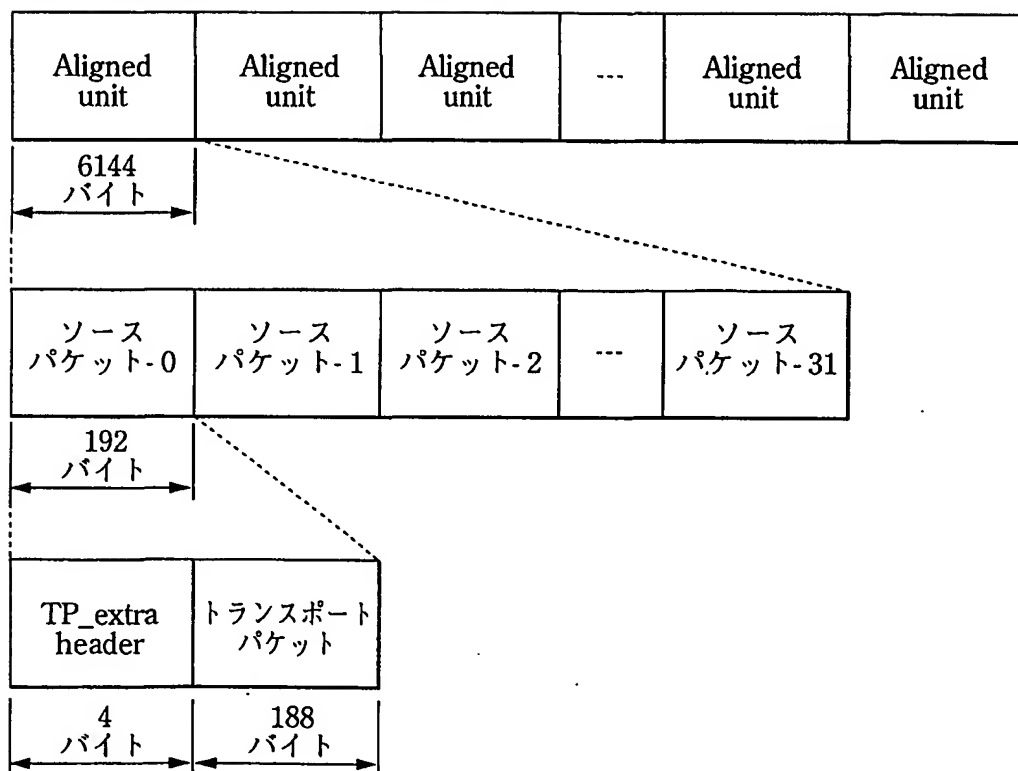


FIG.82

75/100

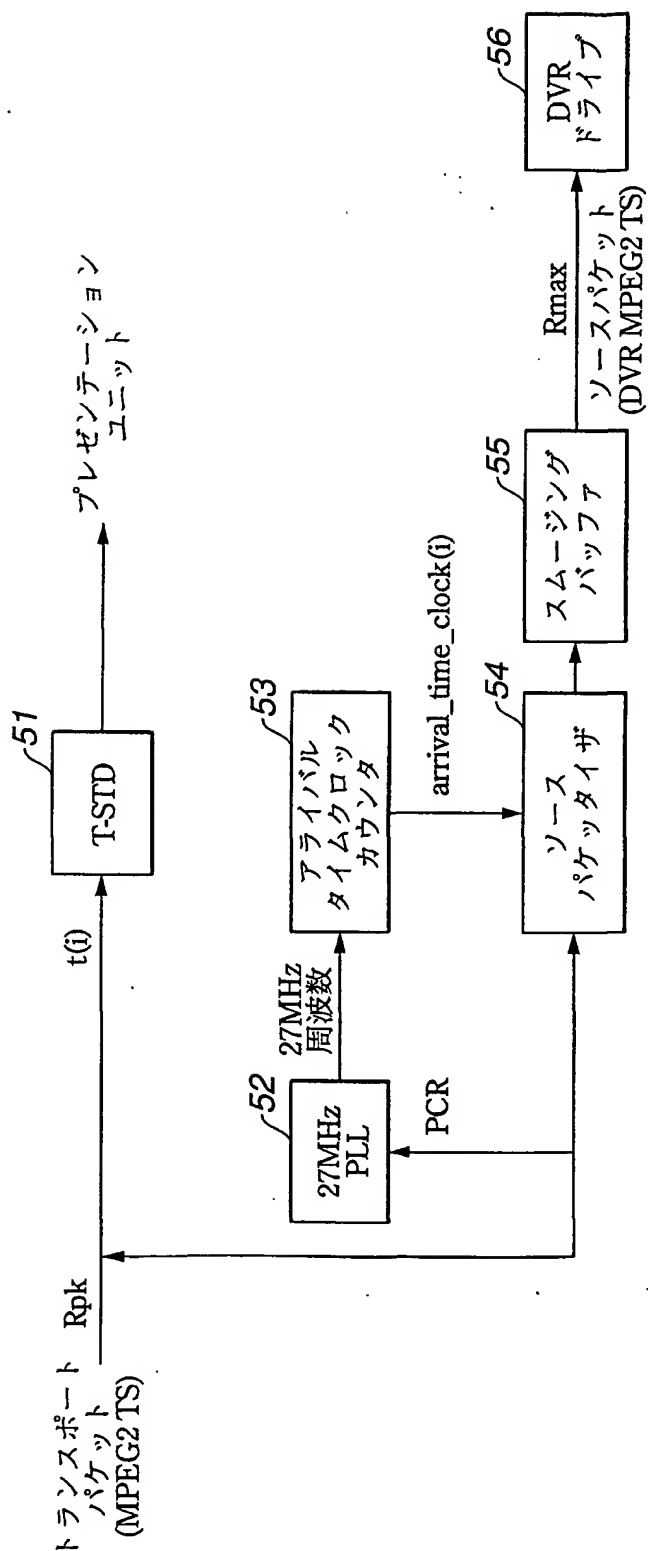


FIG.83

76/100

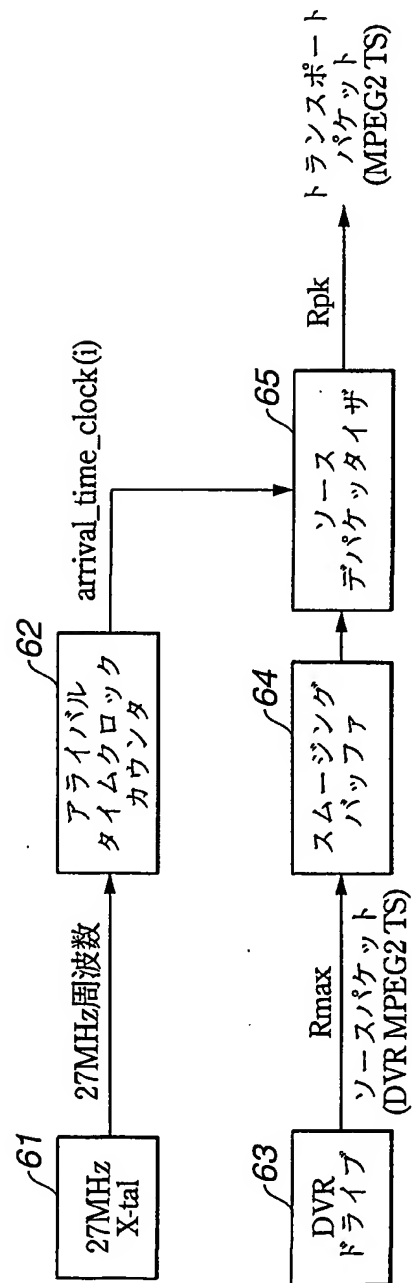


FIG.84

77/100

シンタクス	バイト数	略号
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

FIG.85

78/100

シンタクス	バイト数	略号
TP_extra_header() {		
copy_permission_indicator	2	uimssf
arrival_time_stamp	30	uimssf
}		

FIG.86

79/100

copy_permission _indicator	意味
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.87

80/100

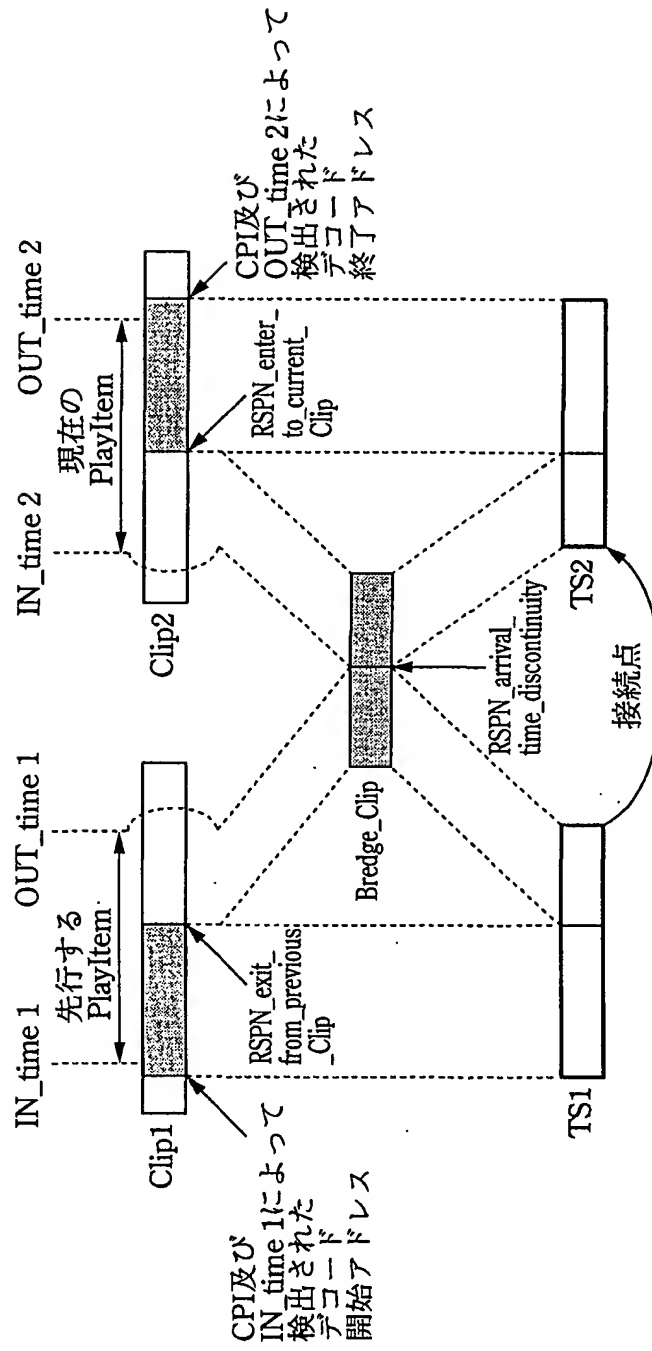


FIG.88

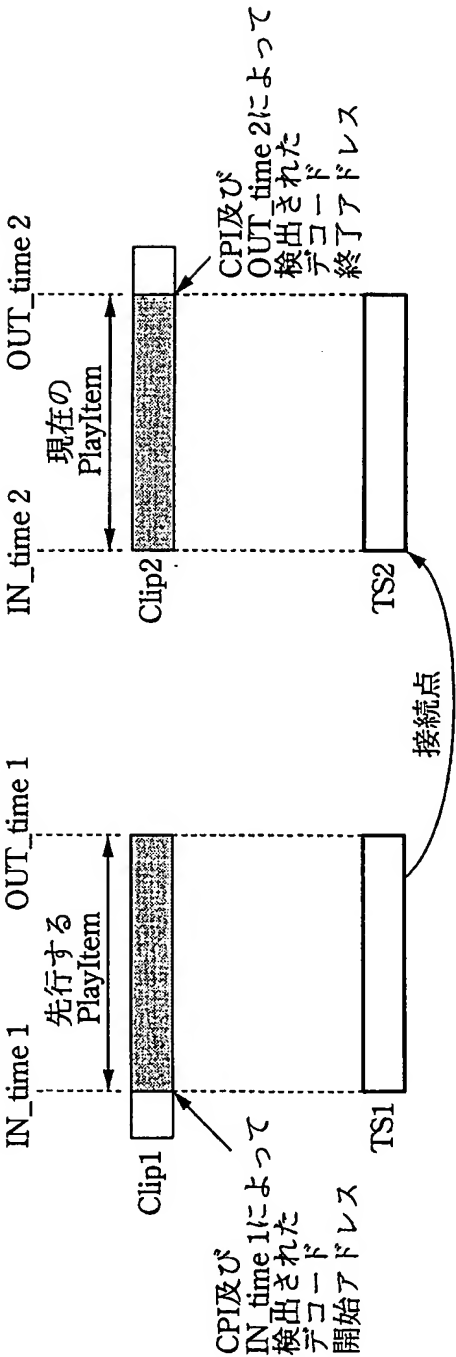


FIG.89

82/100

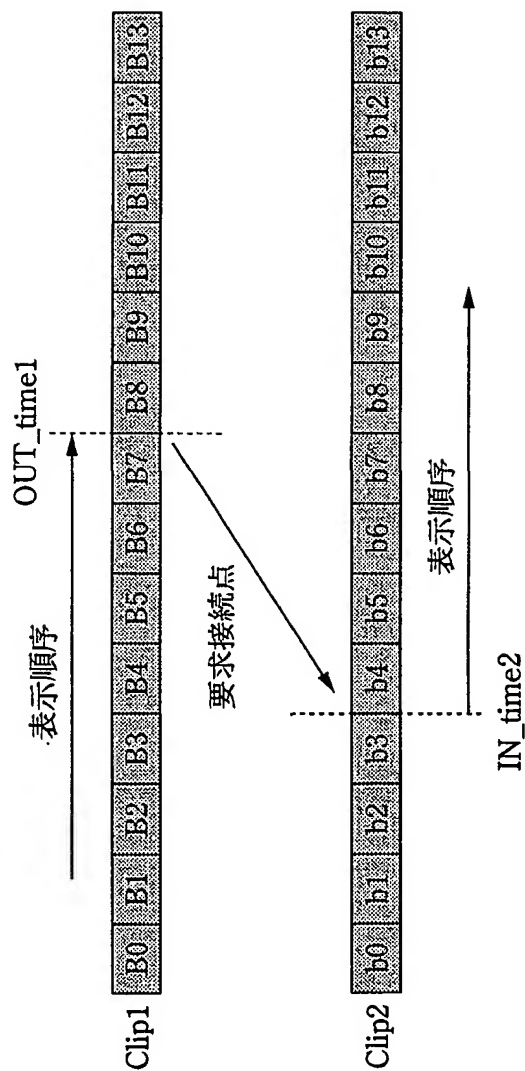


FIG.90

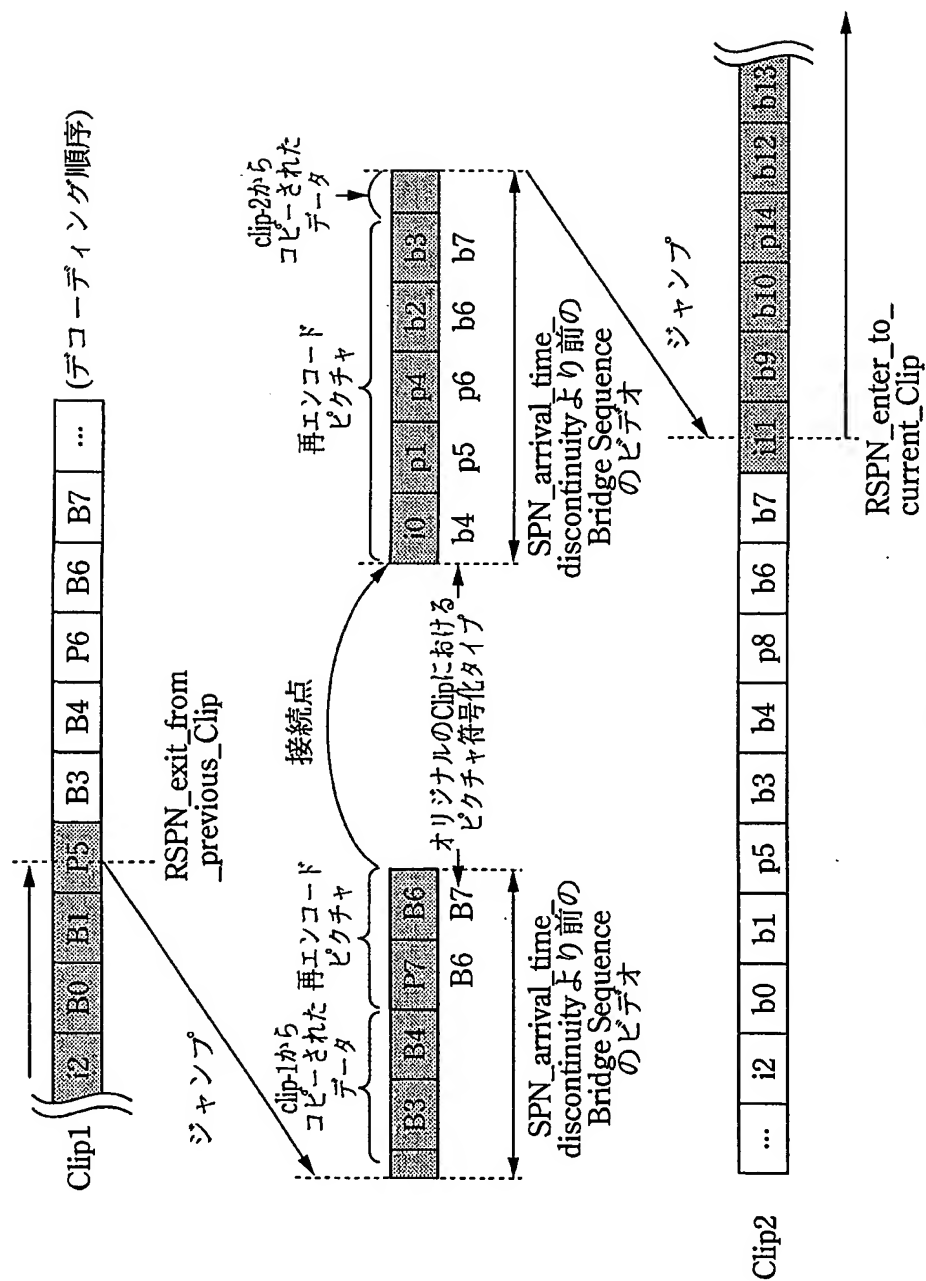


FIG.91

84/100

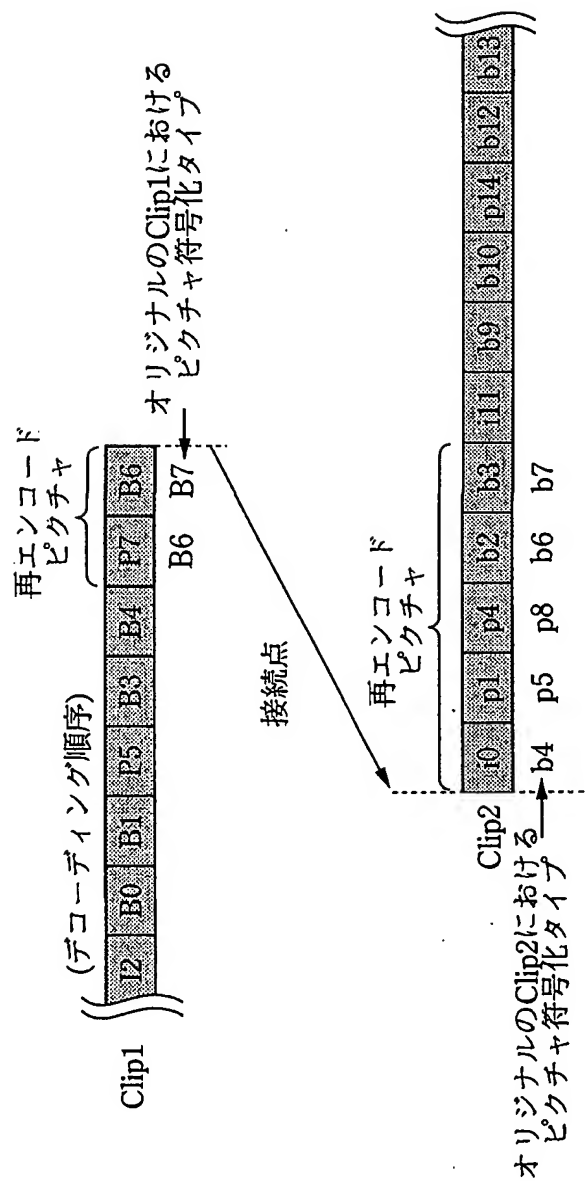


FIG.92

85/100

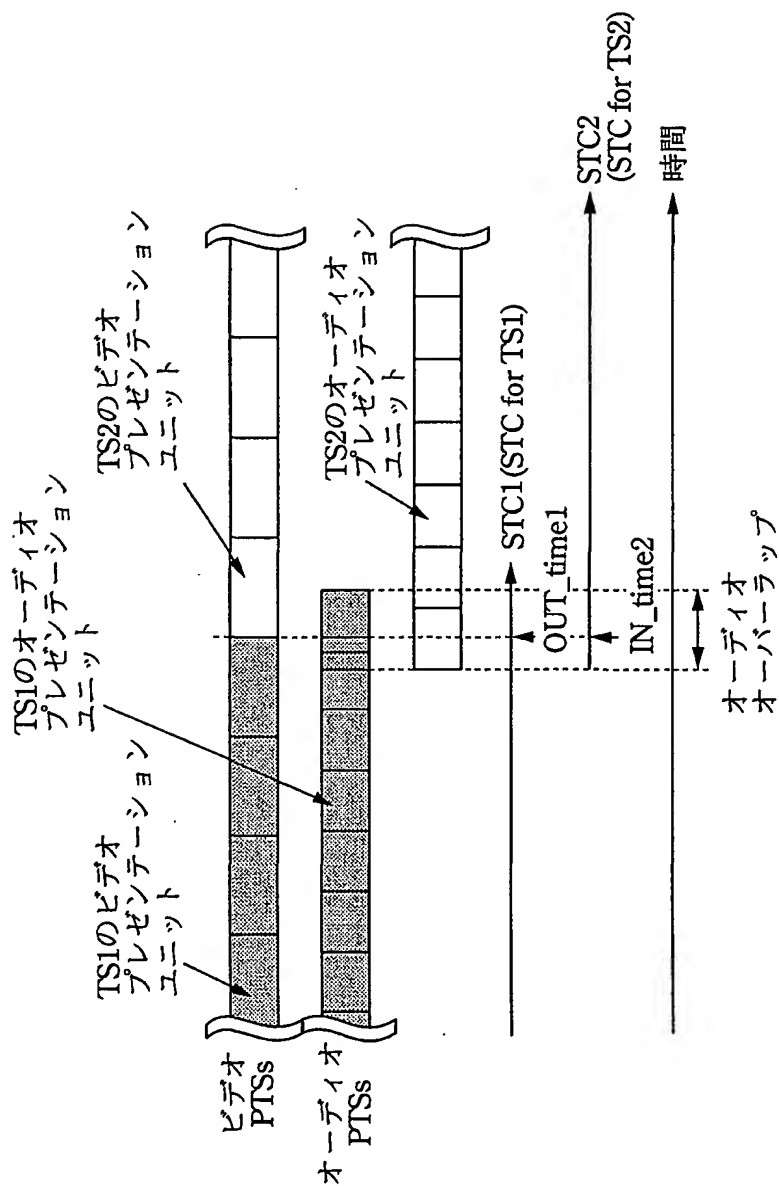


FIG.93

86/100

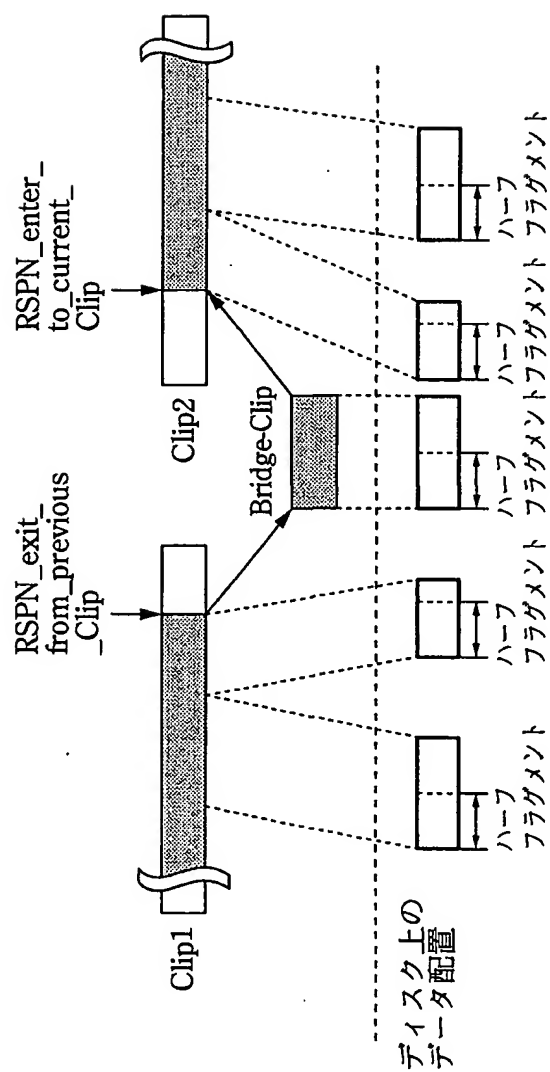


FIG.94

87/100

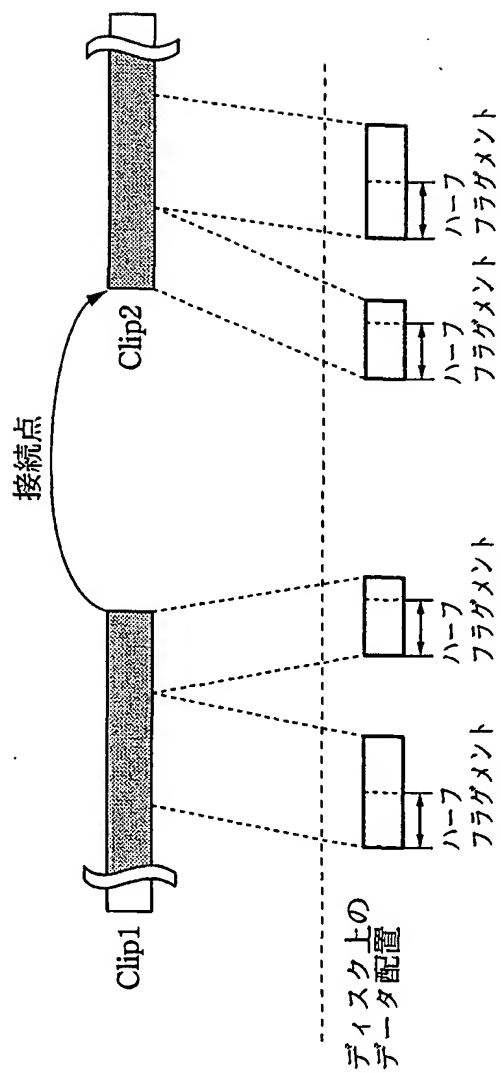


FIG.95

88/100

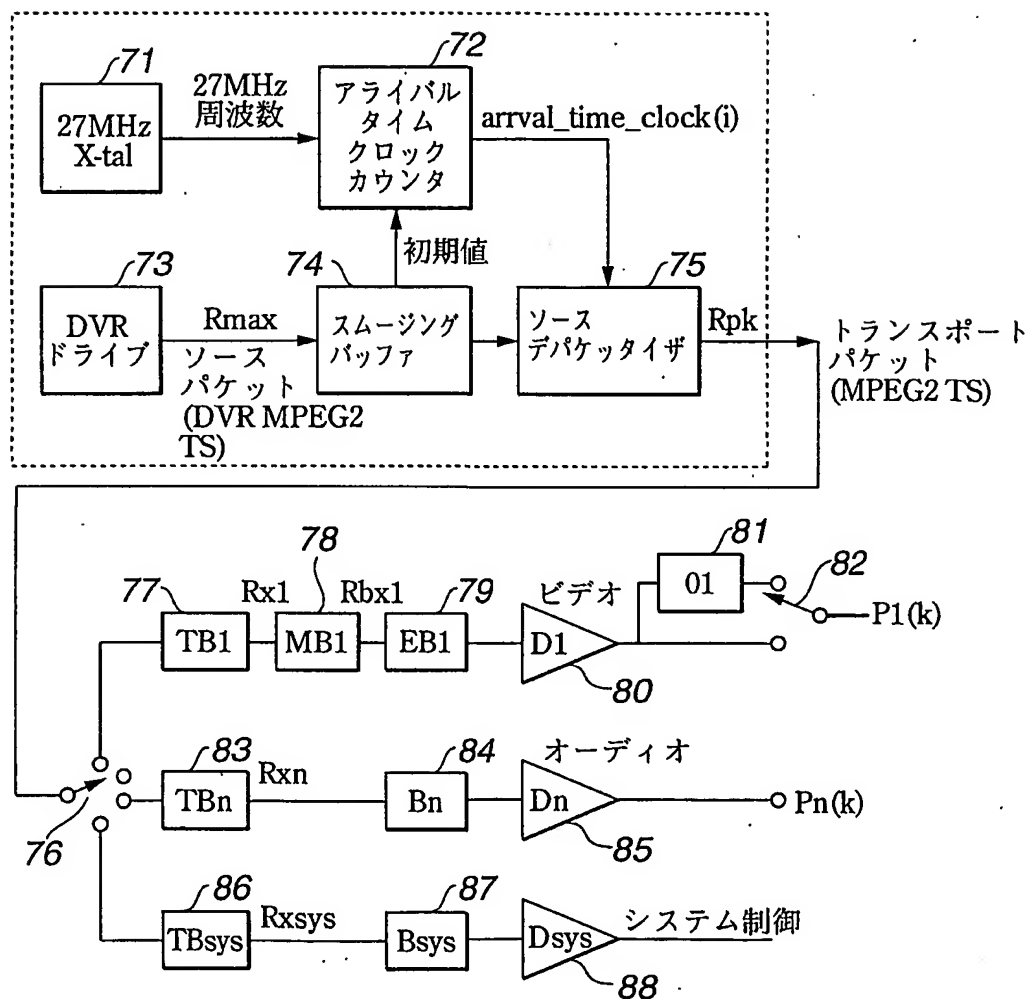


FIG.96

89/100

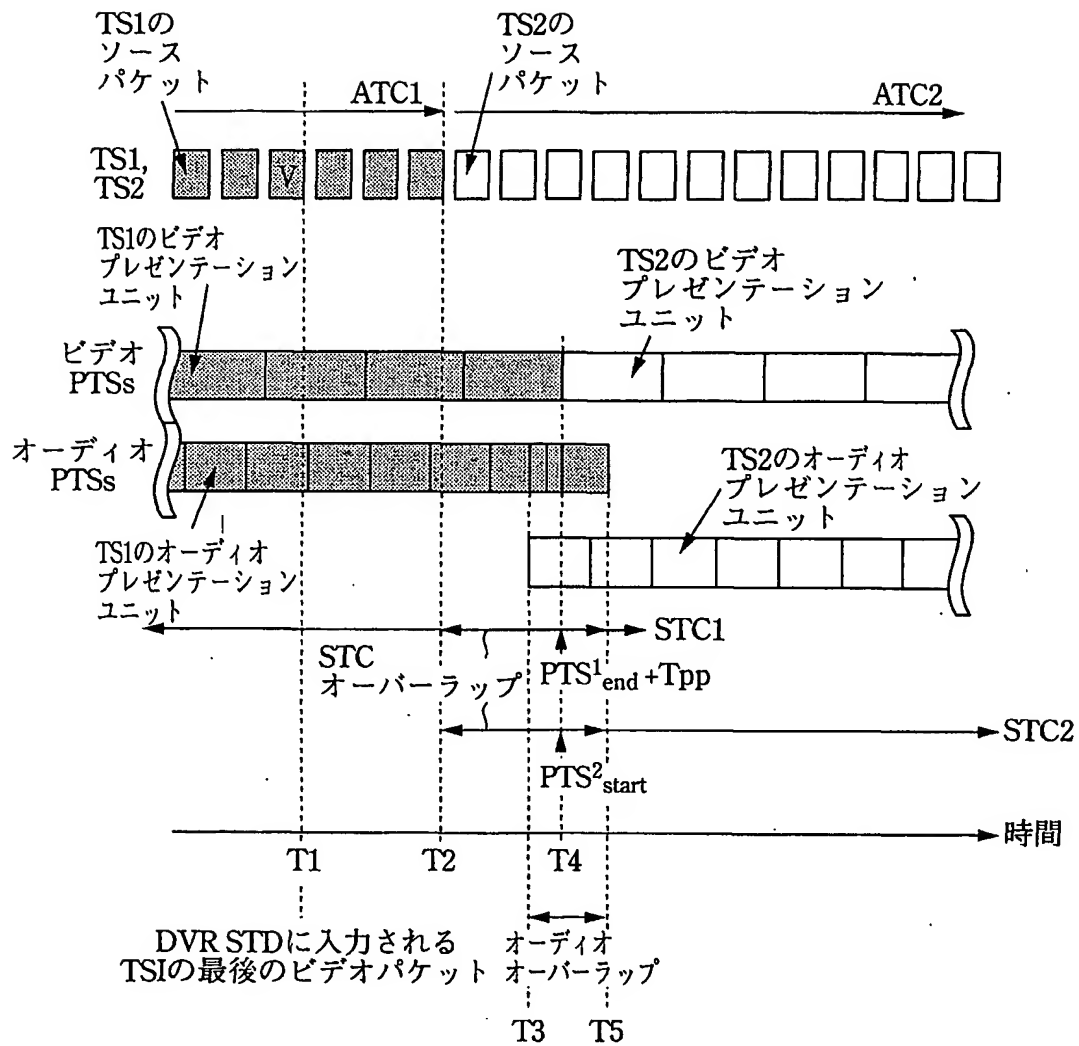


FIG.97

90/100

シンタクス	バイト数	略号
xxxxxx.rpls / yyyyy.vpls {		
version_number	8*4	bslbf
PlayList_start_address	32	uimsbf
PlayListMark_start_address	32	uimsbf
MakersPrivateData_start_address	32	uimsbf
reserved_for_future_use	160	bslbf
UIAppInfoPlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayList()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
}		

FIG.98

91/100

シンタクス	バイト数	略号
UIAppInfoPlayList(){		
length	32	uimsbf
PlayList_service_type		
PlayList_character_set	8	uimsbf
reserved_for_word_align	3	bslbf
playback_control_flag	1	uimsbf
write_protect_flag	1	uimsbf
is_played_flag	1	uimsbf
archive	2	uimsbf
record_time_and_date	4*14	bslbf
duration	4*6	bslbf
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
channel_number	16	uimsbf
reserved_for_word_align	8	bslbf
channel_name_length	8	uimsbf
channel_name	8*20	bslbf
PlayList_name_length	8	uimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	uimsbf
PlayList_detail	8*1200	bslbf
}		

FIG.99

92/100

シンタクス	バイト数	略号
PlayList(){		
length	32	uimsbf
reserved_for_word_align	15	bslbf
CPI_type	1	bslbf
number_of_PlayItems	16	uimsbf
if (<Virtual PlayList> && CPI_type==0){		
number_of_SubPlayItems	16	uimsbf
}else{		
reserved_for_word_align	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem()		
}		
if (<Virtual PlayList>&& CPI_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem()		
}		
}		
}		

FIG.100

93/100

シンタクス	バイト数	略号
SubPlayItem(){		
length	16	uimsbf
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
STC_sequence_id	8	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
sync_PlayItem_id	16	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
}		

FIG.101

94/100

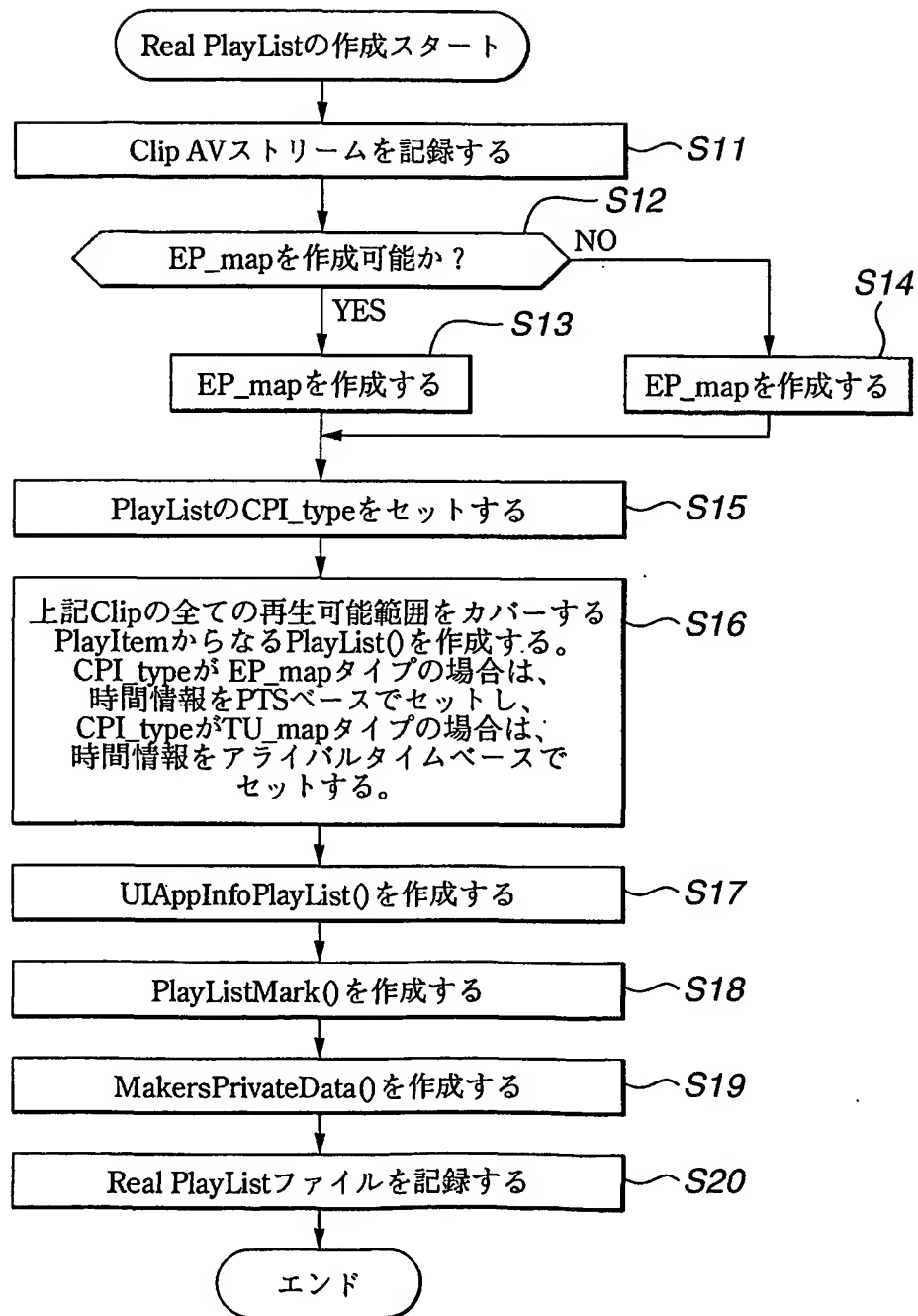


FIG.102

95/100

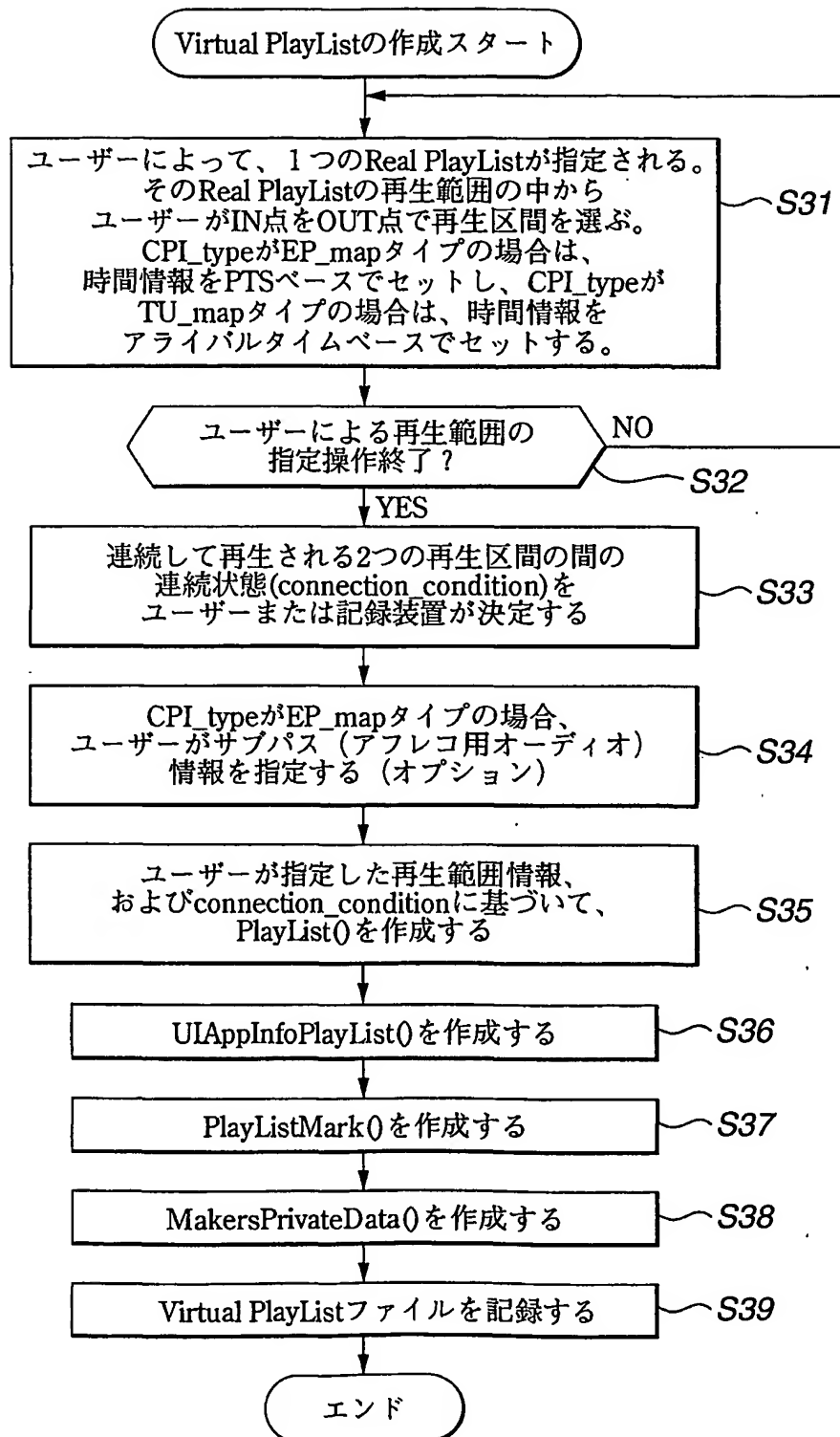


FIG.103

96/100

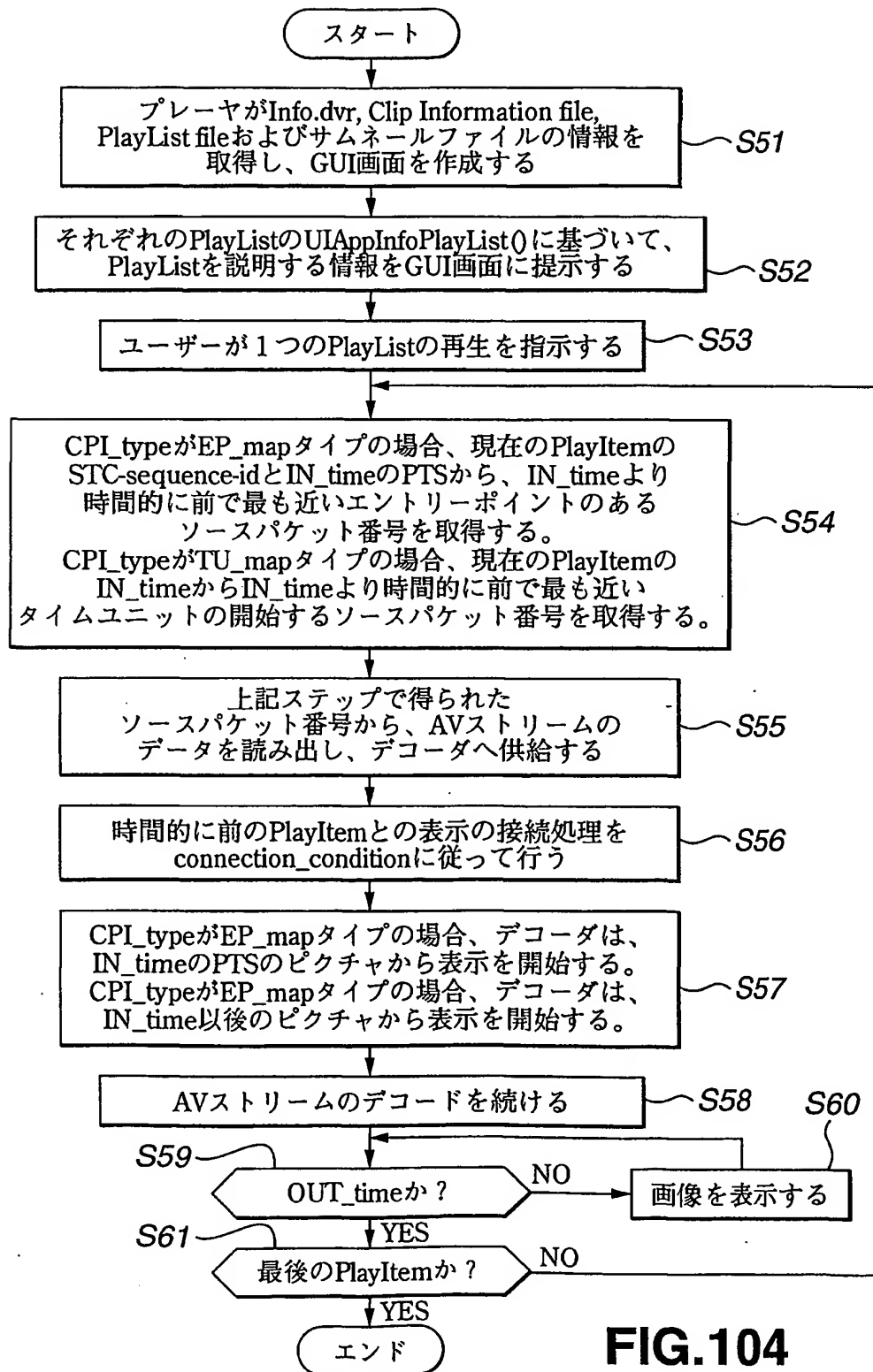


FIG.104

97/100

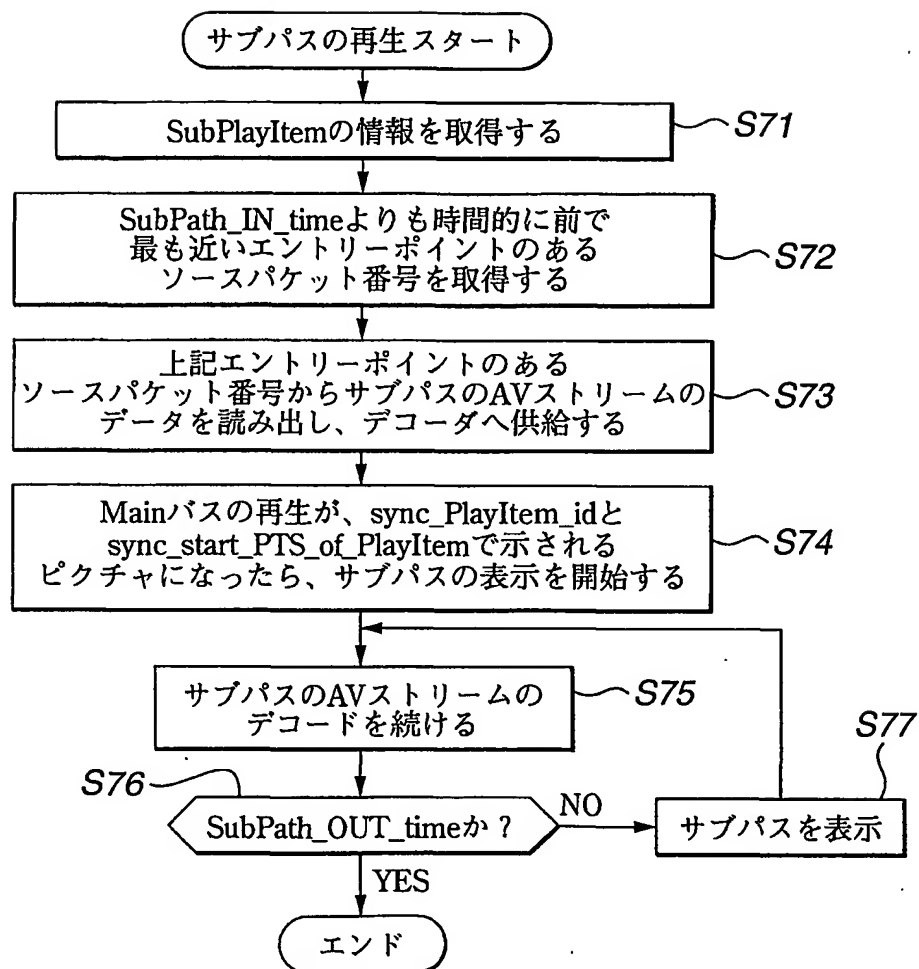


FIG.105

98/100

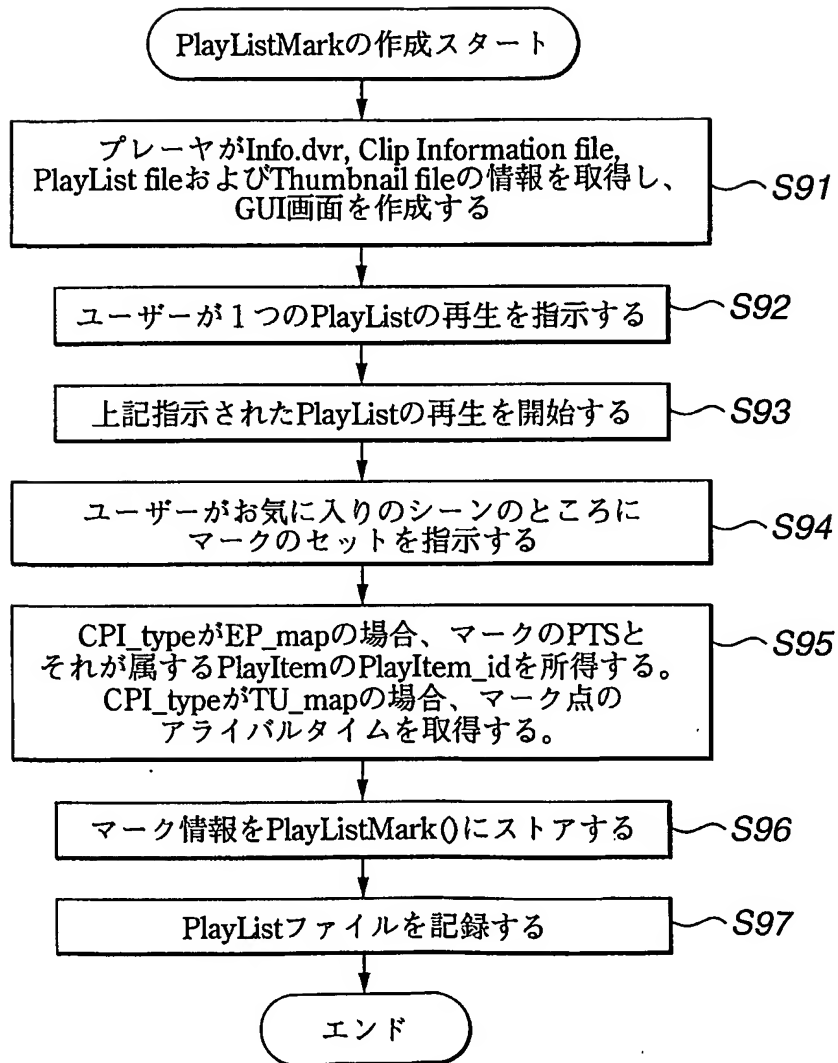


FIG.106

99/100

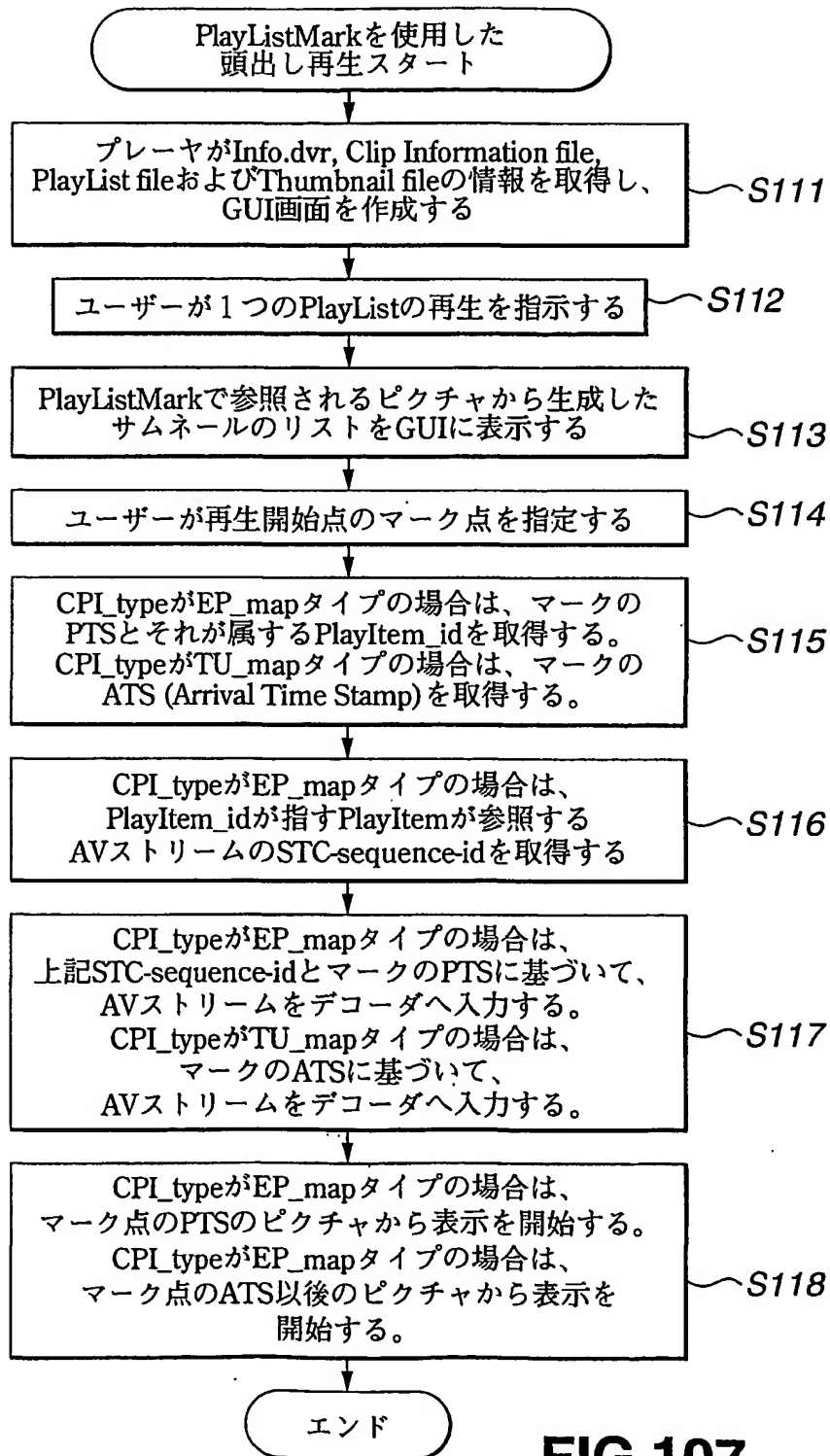


FIG.107

100/100

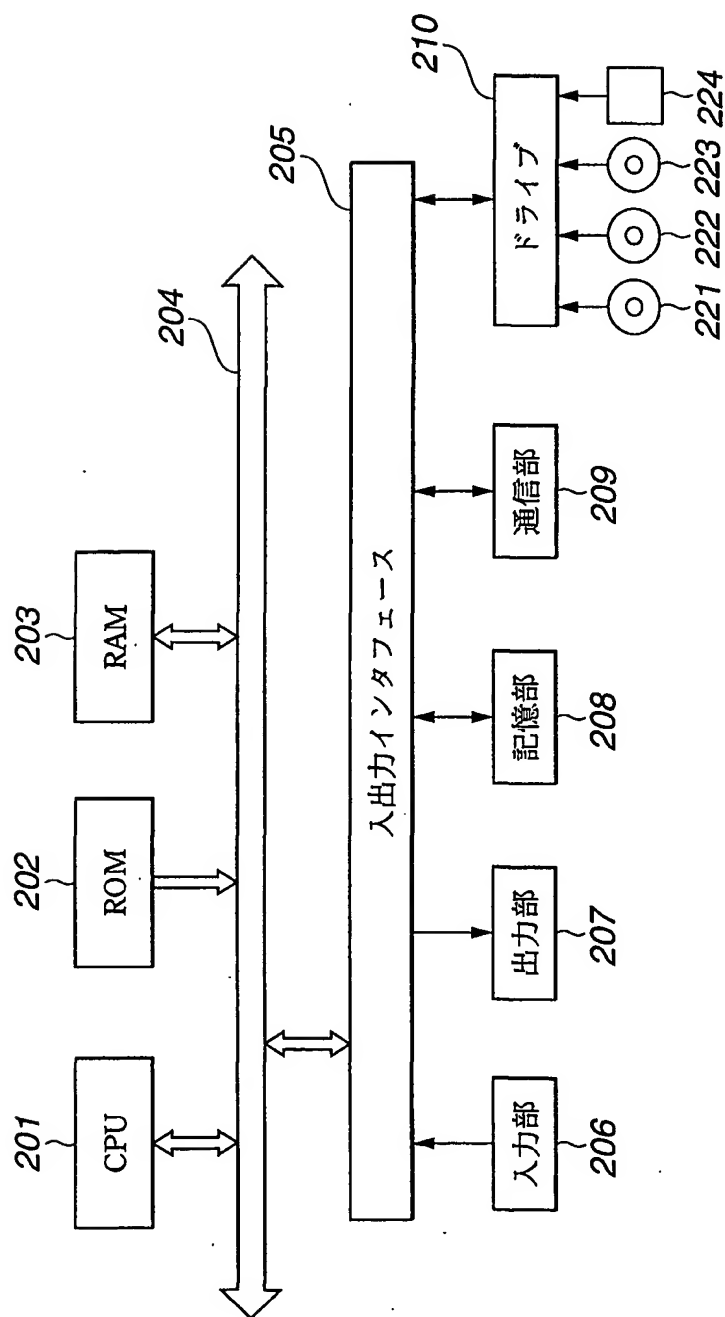


FIG.108

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03417

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl.⁷ H04N 5/93, G11B 20/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl.⁷ H04N 5/76-5/956, 7/24-7/68, G11B 20/10-20/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001
 Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
E, A	JP, 2001-157145, A (Sony Corporation), 08 June, 2001 (08.06.01), Full text; Figs. 1 to 29 (Family: none)	1-14
A	JP, 11-243517, A (Sony Corporation), 07 September, 1999 (07.09.99), Full text; Figs. 1 to 4 (Family: none)	1-14

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

Date of the actual completion of the international search
12 June, 2001 (12.06.01)Date of mailing of the international search report
19 June, 2001 (19.06.01)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))		
Int Cl ⁷ H04N 5/93, G11B 20/10		
B. 調査を行った分野		
調査を行った最小限資料 (国際特許分類 (IPC))		
Int Cl ⁷ H04N 5/76-5/956, 7/24-7/68, G11B 20/10-20/12		
最小限資料以外の資料で調査を行った分野に含まれるもの		
日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2001年 日本国登録実用新案公報 1994-2001年 日本国実用新案登録公報 1996-2001年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
E, A	JP, 2001-157145, A (ソニー株式会社) 08. 6月. 2001 (08. 06. 01) 全文、第1-29図 (ファミリーなし)	1-14
A	JP, 11-243517, A (ソニー株式会社) 07. 9月. 1999 (07. 09. 99) 全文、第1-4図 (ファミリーなし)	1-14
<input type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日	12. 06. 01	国際調査報告の発送日
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 鈴木 明 電話番号 03-3581-1101 内線 3541

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.